N° d'ordre: 2689

THÈSE

présentée

devant l'Université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention INFORMATIQUE

par

Yannick PENCOLÉ

Équipe d'accueil : Dream/Irisa
École Doctorale : Matisse
Composante universitaire : Ifsic

Titre de la thèse :

Diagnostic décentralisé de systèmes à événements discrets : application aux réseaux de télécommunications

Soutenue le le 28 juin 2002 devant la commission d'examen

M. :	Claude	JARD	Président
MM. :	Philippe	DAGUE	Rapporteurs
	Stéphane	LAFORTUNE	
MM. :	Albert	BENVENISTE	Examinateurs
	Marie-Odile	CORDIER	
	Christophe	DOUSSON	
	Patrick	TAILLIBERT	

AVANT-PROPOS

Ça fait maintenant plus de trois années que je travaille sur un thème de recherche précis. Cette expérience je l'ai souhaitée depuis longtemps et ce que j'ai acquis est beaucoup plus enrichissant que ce que j'imaginais, en particulier le contact des gens de différents milieux scientifiques (chercheurs, enseignants, étudiants, industriels), les voyages notamment à l'étranger (États-Unis, Italie, Mexique...). Je me suis mis à étudier le diagnostic un peu par hasard et je me suis rendu compte que les types de raisonnement que l'on utilisait dans ce domaine étaient proches des miens.

Ayant toujours été intéressé par l'étymologie des mots (en particulier quand ils sont d'origine grecque), je me suis penché sur l'étymologie du mot *diagnostiquer* qui en dit déjà long sur ce type de raisonnement : du grec διαγιγνώσκω mot composé de δια (qui signifie la séparation) et de γιγνώσκω qui signifie *apprendre à connaître*. Ainsi, l'origine étymologique de ce mot signifie *apprendre à séparer les connaissances* : séparer le faux du vrai, le bien du mal, ce qui est sain de ce qui est malade, ce qui est normal de ce qui est en panne, autrement dit le sens étymologique du mot *diagnostiquer*, c'est *savoir discerner*.

Si je pouvais résumer mes études doctorales en quelques mots, je dirai qu'elles ont été elle-même un long processus de discernement. Au niveau de mon travail de thèse, j'ai eu la chance de travailler sur des problèmes fondés sur des applications réelles. Ces applications sont complexes, liées à des connaissances, des technologies que l'on ne connaît pas *a priori* et il n'est jamais évident de définir la nature d'un problème intéressant à résoudre sur ce genre d'applications. Dans cette même idée, j'ai toujours été motivé pour faire de l'enseignement au cours de ma thèse et j'ai eu la chance, je crois, de pouvoir enseigner à des étudiants qui découvraient l'informatique. Cette tâche m'a demandé d'avoir un grand recul sur les concepts que j'enseignais et cette prise de recul m'a été profitable dès lors que je revenais sur mes travaux de thèse, si spécifiques et techniques. Cette alternance entre recherche et enseignement a été un travail intellectuel très enrichissant et m'a permis paradoxalement d'avancer plus vite dans ma réflexion de recherche.

Le résultat de cette thèse n'est bien évidemment pas lié qu'à moi seul mais à un ensemble de personnes que je tiens à remercier dans ces quelques lignes. Si je devais les nommer explicitement, la liste serait trop longue aussi je ne vais nommer personne ; dans ce qui suit, chacun se reconnaîtra ! En premier lieu, je voudrais remercier les membres de mon jury pour avoir accepté d'en faire partie. Ils ont tous été pour moi des personnes dont l'influence intellectuelle a été permanente au cours de mes études doctorales. Mention spéciale pour ma directrice de thèse pour m'avoir donné envie de continuer dans cette voie. Durant ces trois années, elle a su me donner ce regard scientifique qui permet de mieux comprendre les choses aussi bien dans la vie professionnelle que privée. Je voudrais également remercier tous les enseignants-chercheurs avec qui j'ai collaboré au cours de mes enseignements à l'Ifsic, pour la confiance qu'ils m'ont accordé et l'expérience qu'ils m'ont fait découvrir. Je remercie également mon collègue de bureau : quand je suis arrivé dans son bureau, j'étais

étudiant en DEA, il était thésard et chef, maintenant que je le quitte, je suis docteur et il est chercheur et ami. Certains diront : « Mais comment Yannick a-t-il fait pour supporter son collègue de bureau ? ». La réponse est : « J'en sais rien, toujours est-il que je vais le quitter avec regret ! ». Merci à toute la bande du patio pour leur sympathie, leur simplicité, leur joie de vivre, les nombreux pots offerts et le lecteur DVD... Merci à toute la bande du midi pour avoir constamment partagé au RU les nombreux repas qui ont lieu au cours d'une thèse, merci pour toutes ces idées, toutes ces histoires alambiquées, tous ces débats, toutes ces bêtises... Je voudrais aussi remercier toute la bande de joyeux chanteurs qui ont bien voulu m'accepter en leur sein avec qui j'ai partagé des moments heureux et chantants. Merci aussi à tous les escrimeurs et à tous les membres de *Bretagnes En Marches 2002*, pour m'avoir permis de m'escrimer encore plus ! Enfin, le plus important, mes remerciements vont à mes parents qui me font confiance, me supportent, et cela depuis ma naissance.

Je vais terminer cet avant-propos par une citation grecque que j'ai toujours eu en tête au cours de mon travail de thèse :

Γνῶτι σεαυτον.

TABLE DES MATIÈRES

INTRODUCTION

Depuis quelques années, les moyens de communications entre les hommes ont subi une révolution : l'apparition puis la démocratisation des réseaux informatiques. Ces réseaux, constituant entre autres l'Internet, forment un outil offrant la possibilité à toute personne d'échanger, de communiquer avec toute autre personne à travers le monde. Cette émergence des réseaux est liée en particulier au développement de technologies pointues dans plusieurs domaines scientifiques (physique, électronique, informatique). Le résultat est la mise en place d'un outil extrêmement complexe qui doit faire face à une utilisation de plus en plus intensive.

Les réseaux de télécommunications font désormais partie de ces systèmes qu'il est crucial de surveiller afin d'assurer leur bon fonctionnement tout au long de leur exploitation. Les ressources informatiques augmentant, on voit apparaître une automatisation des systèmes de surveillance afin d'aider tout opérateur de supervision face à la complexité du travail : les systèmes étant complexes, leurs dysfonctionnements potentiels le sont nécessairement.

Cette thèse s'inscrit dans ce courant d'automatisation puisque son objectif est de fournir une aide intelligente à un opérateur chargé de la surveillance d'un tel système. Étant donné un flot continu d'alarmes reçues par un centre de supervision, l'objectif de ces travaux a été la mise en place d'un système de diagnostic qui est en mesure d'analyser ce flot d'alarmes et d'en donner une interprétation plus compréhensible pour un opérateur humain. Le système supervisé étant réparti et de grande taille, l'idée de cette thèse est d'adopter une *approche décentralisée*. Cette approche s'appuie sur le paradigme bien connu de *diviser pour régner* et sur les techniques de *diagnostic* dites *à base de modèle*. L'un des défis de ce travail est de mettre en place des algorithmes pour le diagnostic de systèmes dynamiques qui répondent à ce problème tout en étant le plus efficace possible. La complexité du problème vient en particulier du fait que les informations à traiter sont nombreuses (taille du système, nombre d'alarmes reçues...) et que l'on cherche à établir une réponse exhaustive : quelles sont les explications possibles du comportement observé ? Quels dysfonctionnements le système subit-il ou a-t-il subi ?

Ce travail a été motivé par la possibilité de traiter des applications réelles de réseaux de télécommunications. L'étude de ces applications a été effectuée dans le cadre de deux projets de recherche. Dans le cadre du projet Gaspar (Gestion d'alarmes par simulation de pannes sur le réseau, projet en collaboration avec France Telecom R&D), l'étude s'est portée sur un réseau de télécommunications en cours d'exploitation : le réseau Transpac. Dans le cadre du projet Magda (Modélisation et apprentissage pour une gestion distribuée des alarmes, projet RNRT), nous avons étudié le cas d'un réseau SDH (hiérarchie numérique synchrone).

L'organisation du document est la suivante. Dans un premier temps, le cadre de la gestion des réseaux de télécommunications est présenté. Ce premier chapitre a pour but de décrire les différentes difficultés liées à la surveillance et au diagnostic de pannes dans ces systèmes et d'en dégager les besoins. Le deuxième chapitre sera consacré à une discussion autour des approches déjà étudiées pour faire face à ce problème. Les travaux de recherche sur le diagnostic

de pannes dans les systèmes dynamiques sont très nombreux. Les approches étudiées sont très diverses et chacune est révélatrice de son époque. Ces approches dépendent essentiellement des technologies utilisables et des ressources informatiques disponibles au moment de leur développement. Dans le troisième chapitre, nous présentons les principes fondamentaux de notre contribution. Ces principes constituent un cadre formel sur lequel se fondent l'approche étudiée et sa mise en œuvre. Les quatrième et cinquième chapitres présentent les choix algorithmiques en vue de l'implémentation d'une application informatique de l'approche étudiée. Le quatrième chapitre s'oriente sur la description des algorithmes de base permettant d'établir un diagnostic en fonction d'un ensemble d'observations. Le cinquième chapitre présente les aspects liés au caractère en ligne du diagnostic. Dans ce contexte, les observations ne sont pas toutes connues *a priori*, une adaptation du diagnostic prenant en compte de nouvelles observations est nécessaire, cette adaptation étant effectuée à l'aide d'un algorithme *incrémental* de diagnostic. Le dernier chapitre présente l'application *Ddyp*. Cette plate-forme a été développée tout au long de cette thèse afin d'exploiter et de valider les différents aspects de l'approche étudiée. Nous présentons également dans ce chapitre le déploiement de cette plate-forme sur deux exemples de réseaux issus de cas réels : le réseau Transpac et le réseau de type SDH.

CHAPITRE *1*

Diagnostic de pannes dans les réseaux de télécommunications

1.1 Problématique

Les réseaux de télécommunications ont pris un essor important dans le monde. Aujourd'hui, ce sont des systèmes que nous utilisons quotidiennement (utilisation de l'Internet par exemple). Les besoins se multipliant, la complexité de ces systèmes augmente. On demande aux réseaux de télécommunications d'être efficaces, robustes, sûrs et toujours disponibles. Leur gestion devient une activité qui est elle-même de plus en plus complexe. De nouveaux besoins informatiques sont nécessaires afin d'améliorer la gestion par son automatisation. En particulier, l'un des points cruciaux de la gestion de réseau, est le diagnostic de pannes. L'objectif est de détecter les problèmes au plus tôt afin de pouvoir assurer la qualité de service demandée par les utilisateurs (sûreté, efficacité, disponibilité). Ce chapitre décrit dans un premier temps un état sur la nature de la gestion de réseau telle que les organismes de normalisation la conçoivent. La deuxième partie de ce chapitre décrit de façon plus détaillée la gestion de pannes, ses objectifs, ses besoins.

1.2 Réseaux de télécommunications

1.2.1 Réseaux informatiques

Les réseaux informatiques sont nés du besoin de faire communiquer des terminaux distants avec un site central, puis des ordinateurs entre eux, tels que des stations de travail avec leurs serveurs. Différentes catégories de réseaux peuvent être dénombrées. On en compte généralement cinq, différenciées par la distance maximale entre les deux points les plus éloignés [Pujolle 95]. Par ordre croissant sur la distance maximale, on distingue les types de réseaux suivants.

1. Les *bus* relient les processeurs, les mémoires, les entrées-sorties d'un calculateur ou d'un multiprocesseur.
2. Les *structures d'interconnexions* relient dans une même pièce, ou à des distances faibles, différents calculateurs entre eux.

3. Les *réseaux locaux* que l'on appelle aussi LAN (*Local Area Network*) correspondent par leur taille aux réseaux intra-entreprise. Leur objectif est de permettre le transport de toutes les informations numériques localement.
4. Les *réseaux métropolitains* ou MAN (*Metropolitan Area Network*) correspondent à une interconnexion de plusieurs bâtiments situés dans une même ville. Ces réseaux doivent être capables d'interconnecter les réseaux locaux des différents bâtiments et de prendre en charge les machines communes à l'ensemble de la gestion du site distribué.
5. Enfin, les *réseaux étendus* ou WAN (*Wide Area Network*) sont destinés, comme le nom l'indique, à transporter des données numériques sur des distances à l'échelle d'un pays ou même de plusieurs.

1.2.2 Réseaux de télécommunications

Les réseaux de télécommunications qui nous concernent sont des réseaux étendus (type WAN). Ces réseaux sont mis à la disposition de tierces personnes. Tout un chacun ayant des besoins de transmission de données électroniques peut louer, selon des modalités d'abonnement, des services de transports (voir la figure 1.1). De la même façon qu'on accède au réseau téléphonique commuté pour transmettre l'information vocale entre deux personnes, on peut utiliser, par exemple, le réseau Télépac en Suisse et le réseau Transpac en France (ou les deux) pour faire dialoguer des ordinateurs. Actuellement les grands réseaux développés par les entreprises s'appuient sur ce type de réseau.

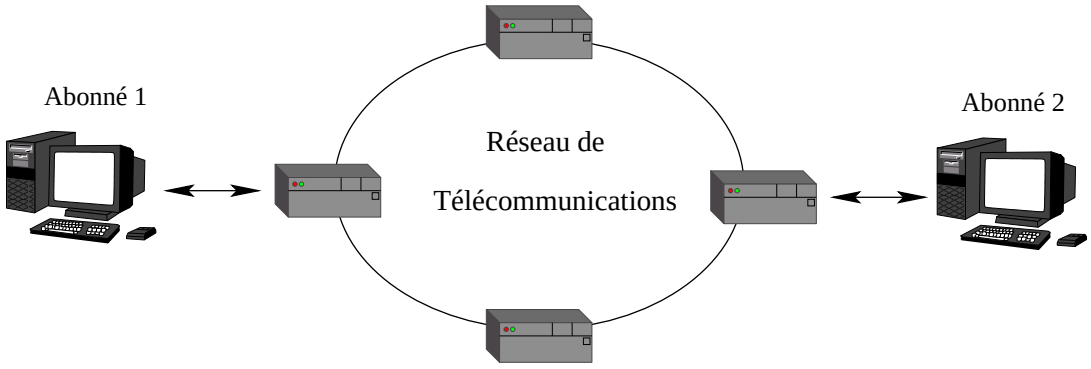

FIG. 1.1 – Réseau de télécommunications.

1.3 Gestion de réseau

1.3.1 Objectifs de la gestion de réseau

Un réseau de télécommunications est un système complexe qu'on doit organiser et gérer. La multiplicité, l'hétérogénéité de ses éléments constitutifs, de ses modes de fonctionnement et de ses applications, rendent sa gestion et sa maîtrise difficile. Les outils et les méthodes de

gestion doivent être conçus et mis en place, si possible lors de la conception du réseau, pour garantir un service de réseau harmonieux et efficace [Arpège 92].

Le réseau requiert un pilotage en souplesse adapté à sa structure dynamique, sujette à des modifications fréquentes. De plus, il est vulnérable, menacé de pannes, de dysfonctionnements et d'utilisation anarchique ou intempestive de ses ressources. La vulnérabilité et la complexité d'un réseau s'accroissent avec sa taille et les niveaux d'hétérogénéité qu'il sous-tend.

Le gestionnaire de réseaux, aussi appelé *Opérateur* [1], doit gérer l'urgence et anticiper l'avenir. Il doit être réactif et préventif, trouver l'équilibre, pour le dimensionnement du réseau, entre les ressources à gérer et les ressources pour gérer, faire un compromis entre les performances et la qualité de service. La gestion de réseau est à prendre en compte à la conception du réseau. Elle doit être modulaire et doit pouvoir se mettre en œuvre aussi bien en local qu'à distance. [Sloman 89] résume ainsi les objectifs de la gestion :

> La gestion d'un système englobe l'ensemble des moyens mis en œuvre pour offrir aux utilisateurs un service de qualité et permettre l'évolution du système en incluant de nouvelles fonctionnalités. Elle vise à optimiser les performances des services pour les utilisateurs et à permettre une utilisation maximale des ressources à un coût minimal.

La gestion de réseau consiste donc à résoudre un ensemble de problèmes fondés sur des ressources communes. [Simony et Znaty 97] présente une séparation des problèmes selon leur nature par l'utilisation de cinq modèles conceptuels (voir figure 1.2).

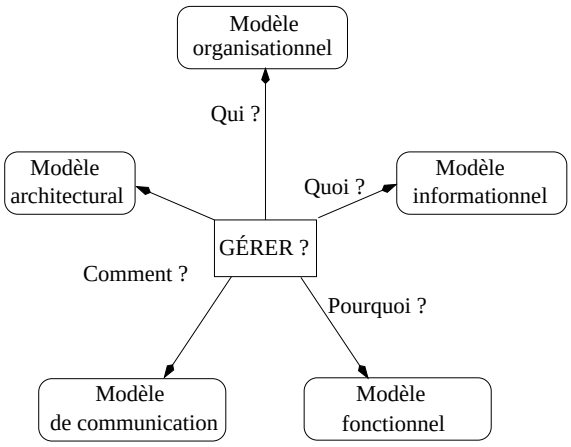

FIG. 1.2 – Les cinq modèles conceptuels de la gestion de réseau.

Chaque modèle répond à une question sur la gestion. Le *modèle informationnel* sert à l'identification et à la représentation des éléments à gérer (le *Quoi ?*). Le *modèle architectural* et le *modèle de communication* décrivent la structure des entités gérées ainsi que la façon dont les outils de gestion peuvent interagir avec ces entités (le *Comment ?*). Le modèle *fonc-*

[1] Dans ce mémoire, on emploiera le terme « opérateur » avec une minuscule pour désigner une personne physique contribuant à la gestion d'un réseau ; et avec une majuscule, pour désigner une personne morale (une organisation) gérant un réseau.

tionnel définit les différentes tâches à effectuer par la gestion (le *Pourquoi ?*) et enfin le *modèle organisationnel* décrit les participants de cette gestion et la façon dont leur sont affectées les différentes fonctions de la gestion.

1.3.2 Modèle informationnel

Le *modèle informationnel* sert à l'identification et à la représentation des éléments à gérer : c'est sur ce modèle que se fonde toute l'infrastructure de la gestion. Il constitue la brique de base sur laquelle s'appuient tous les autres modèles conceptuels de la gestion. L'assurance d'une cohérence fonctionnelle du système ne peut reposer que sur un état informationnel cohérent et sur la mise en œuvre d'une gestion efficace de ces informations.

La norme M.3100 de l'UIT-T [UT 95a] (Union Internationale des Télécommunications-standardisation du secteur des Télécommunications) (anciennement CCITT , Comité Consultatif International Téléphonique et Télégraphique) définit la notion d'*objet géré* comme une vue abstraite d'une ressource physique ou logique de traitement de communications de données. Un tel objet possède des caractéristiques : attributs, actions, notifications, comportements. Les *attributs* sont des propriétés descriptives (type, nom de l'instance représentée, état de fonctionnement...). Les *actions* correspondent aux opérations de gestions qui peuvent être appliquées à ces objets. Les *notifications* sont les informations qui peuvent être émises par l'objet concernant les modifications subies par cet objet. Enfin, le *comportement* spécifie les caractéristiques dynamiques de l'objet, les circonstances dans lesquelles les notifications doivent être émises et les actions appliquées ; il inclut aussi la sémantique de certains attributs et décrit la façon dont les opérations de gestion affectent l'objet et ses attributs.

Ces objets gérés constituent la *base d'information de gestion* (MIB pour *Management Information Base*). Cette base de données est indispensable pour les tâches de gestion parce qu'elle décrit le réseau et les éléments du réseau tels que le système de gestion les perçoit localement. Les protocoles de gestion sont utilisés pour la consulter et pour en effectuer la mise à jour.

1.3.3 Modèle architectural et modèle de communication

L'évolution structurelle des réseaux de télécommunications va actuellement dans le sens d'une claire séparation fonctionnelle des services de communication et de gestion. Il s'agit de faire inter-fonctionner ces deux domaines de service au travers d'interfaces de communication normalisées, et d'intégrer harmonieusement les divers systèmes de gestion par l'emploi d'outils et de méthodes normalisés. L'institut de normalisation UIT-T a élaboré le concept de réseaux de gestion des télécommunications (TMN) (*Telecommunication Management Network*) pour définir une architecture fonctionnelle d'un système de gestion de réseau souple, complet et évolutif (norme M.3010 [UT 95b]). Le réseau de gestion des télécommunications offre un cadre modulaire de développement de la gestion dans lequel les opérateurs, les applications et les équipements de télécommunications communiquent de façon normalisée. Un des points importants de ce cadre architectural est la définition claire des responsabilités de chaque acteur. Le réseau de gestion des télécommunications est fonctionnellement distinct du réseau de télécommunications qu'il gère, interroge et commande, même s'il peut utiliser dans la pra-

tique les ressources de ce dernier. Il est logiquement séparé de ce réseau qui peut être dédié à l'acheminement de la voix, des données ou de l'image (voir figure 1.3).

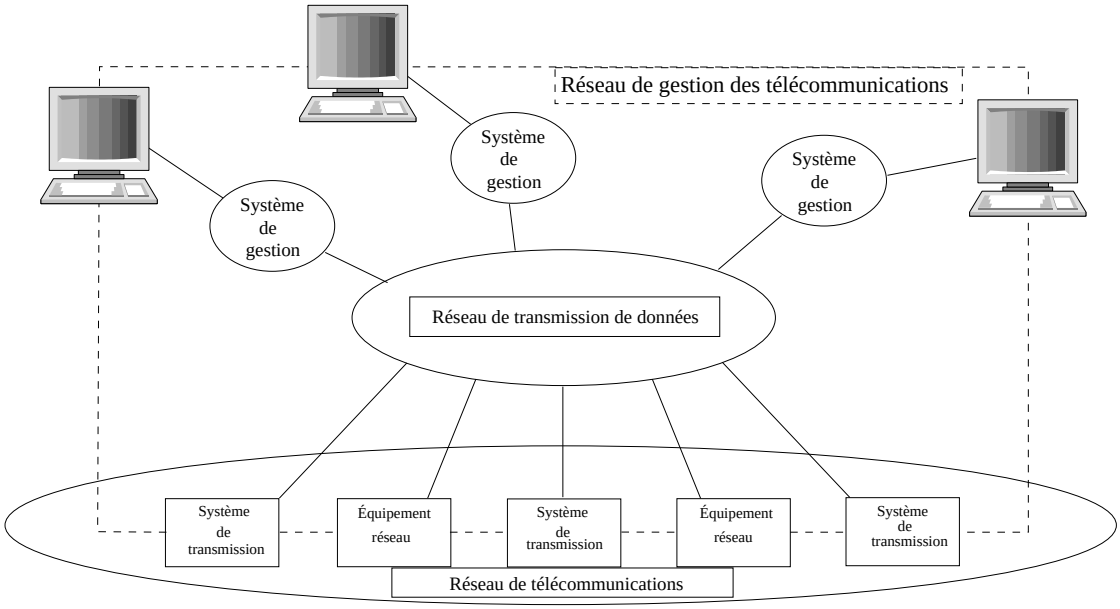

FIG. 1.3 – Relation générale entre le réseau de télécommunications et le TMN.

Les composants d'un TMN sont entre autres :
– *l'élément de réseau* (NE pour *Network Element*), il se compose d'un équipement de télécommunications (un groupe ou une partie) et d'un équipement de support exécutant des fonctions de gestion mais considéré comme faisant partie du réseau de télécommunications ;
– *le réseau de transmission de données* (DCN pour *Data Communication Network*), il s'agit d'un réseau de communication à l'intérieur du TMN pour les échanges d'information de gestion entre les composants du TMN ;
– *le système de gestion* (OS pour *Operation System*), c'est un centre d'administration qui offre des applications de gestion ;
– *la station de travail* (WS pour *Work Station*), elle assure la communication entre l'opérateur humain et les composants du TMN.

1.3.4 Modèle organisationnel

1.3.4.1 Relation gestionnaire/agent

Le concept de base est la relation *gestionnaire/agent*. Le gestionnaire et l'agent sont des processus qui échangent des informations de gestion à travers un protocole de communication. Le gestionnaire initialise les demandes d'opérations sur les objets gérés et reçoit les notifications. L'agent effectue les opérations sur les objets gérés et peut retransmettre les notifications envoyées par les objets au gestionnaire. Chaque agent gère sa propre base d'informations de

gestion sur laquelle le gestionnaire peut travailler. Selon le contexte de communication, un gestionnaire peut avoir un rôle d'agent vis-à-vis d'autres gestionnaires ou agents.

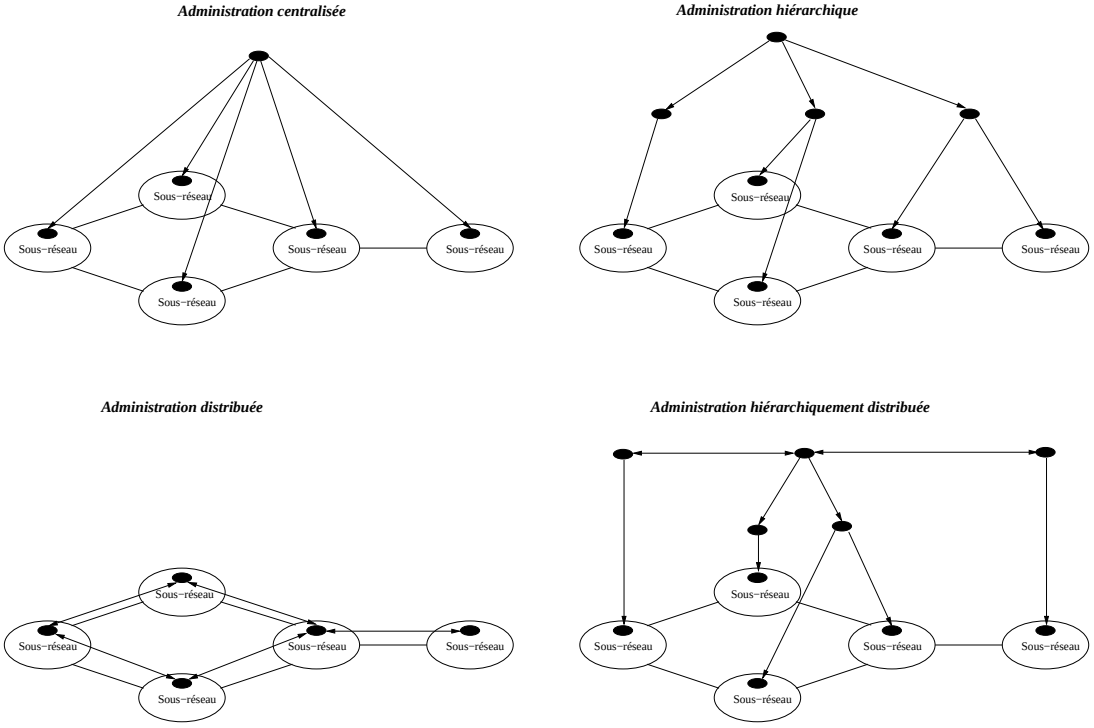

FIG. 1.4 – Organisations possibles de la gestion de réseau.

1.3.4.2 Types d'organisation

La figure 1.4 présente quatre organisations de la gestion de réseau souvent utilisées en pratique.

1. *Administration centralisée*. Un gestionnaire est associé à chaque sous-réseau, chaque gestionnaire joue le rôle d'un agent face au gestionnaire central. Cette organisation présente l'avantage d'être simple, néanmoins, si le nombre de sous-réseaux est important, le gestionnaire central risque d'être surchargé si de multiples opérations de gestion surviennent pendant une période donnée.
2. *Administration hiérarchique*. Cette architecture permet de décharger le gestionnaire central par l'ajout de plusieurs niveaux de gestionnaires intermédiaires. L'organisation hiérarchique permet également de séparer les domaines de responsabilité (par exemple réseau privé/public, national/régional).
3. *Administration distribuée*. Cette organisation est à l'opposé des deux précédentes. Le gestionnaire d'un sous-réseau coopère avec les autres en vue d'obtenir les informations nécessaires sur le voisinage pour la gestion de son sous-réseau. Chaque gestionnaire sert

donc de point d'échanges d'information pour supporter une administration répartie et transparente du réseau. L'avantage de ce genre d'organisation réside dans la répartition totale de la charge de gestion. Par contre, la mise en œuvre du protocole d'échanges entre les gestionnaires peut s'avérer complexe (mise en place d'un système de synchronisation des gestionnaires, d'un système de consensus sur les opérations de gestion à effectuer localement afin d'obtenir l'effet global attendu). De plus, cette organisation peut conduire à une surcharge de trafic sur le réseau de communication de gestion.

4. *Administration hiérarchiquement distribuée*. Cette organisation apporte à la précédente la notion de hiérarchie de gestionnaires. Seuls les gestionnaires de plus haut niveau servent de points d'échanges pour l'administration répartie. Cette organisation permet de diminuer le trafic sur le réseau de communication de gestion. De plus, cette vision hiérarchique permet de séparer la gestion par domaine de responsabilités, ce qui est mieux adapté pour la prise des décisions locales.

1.3.4.3 Centre de supervision

Le *centre de supervision* est un centre où sont exploités les systèmes de gestion par des opérateurs humains. Il constitue une partie de la mise en œuvre du réseau de gestion des télécommunications (voir section 1.3.3) : l'interface homme-machine. Les fonctions d'un tel centre sont :

- de recevoir les informations (en temps réel) sur la qualité de service et la qualité de fonctionnement ;
- d'aiguiller ces informations vers des opérateurs, applications ou bases de données ;
- de traiter les données brutes afin d'offrir à l'opérateur un synoptique de l'état du réseau et de l'avertir ;
- de présenter ces informations de façon ergonomique à l'opérateur (par exemple, les centres de supervision sont munis d'un mur d'écrans spectaculaire présentant une vue géographique du réseau) ;
- de permettre aux opérateurs d'agir à distance sur le réseau (activation de plans de secours par exemple) ;
- de mémoriser les interventions des opérateurs afin de les analyser et d'améliorer les procédures d'interventions.

1.3.5 Modèle fonctionnel

Les objectifs principaux de la gestion de réseau en télécommunications peuvent être décomposés en cinq aires fonctionnelles.

1. *La gestion des pannes* : détection des dysfonctionnements anormaux, signalisation d'alarmes, localisation des pannes, des réparations, confirmation du retour en fonctionnement normal.
2. *La gestion de la configuration* : initialisation et lancement du réseau, contrôle et présentation de l'état du système, établissement de l'historique.
3. *La gestion des performances* : évaluation du temps de réponse, du débit, du taux d'erreur, de la disponibilité.

4. *La gestion des informations comptables* : définition des coûts d'utilisation et des taux pour chaque ressource facturable, du coût global au niveau du réseau ou d'une application particulière.
5. *La gestion de la sécurité* : la mise en œuvre des politiques de sécurité assurant le contrôle d'accès, l'authentification des correspondants, la confidentialité et l'intégrité des données.

Les aires fonctionnelles ne sont pas indépendantes ou disjointes, mais peuvent se reposer sur des mécanismes et des informations communes. Par exemple, certaines mesures de qualité de service appliquées dans la fonction de la gestion des performances peuvent être également utilisées pour la gestion des pannes. De même, les activités de gestion de la configuration sont fortement corrélées avec celles de la gestion des pannes ; d'une part, la connaissance des configurations actuelles et de l'historique peut être indispensable pour la localisation des pannes, d'autre part, la réparation des pannes peut être réalisée par la reconfiguration de certaines ressources du réseau.

1.3.6 Conclusion

Avec la multiplicité des équipements de réseaux et la complexité des environnements souvent hétérogènes, la quantité d'informations à traiter (événements, actions de routage, configuration, etc) devient considérable. À une panne peut correspondre un ensemble d'éléments dont la corrélation renseigne sur la nature et sur l'origine de la panne. À un mauvais temps de réponse peuvent être associés des routages mal adaptés et des ressources mal calibrées (lignes de capacité trop faible, mémoire des contrôleurs insuffisante, etc).

En s'appuyant sur ces exemples, en supposant qu'un centre de supervision reçoive toutes ces informations, il est confronté à certaines difficultés : savoir interpréter les informations qui lui parviennent de façon à surveiller et réagir efficacement, et ne pas trop solliciter (à des fins de gestion) les lignes et les éléments de réseaux (dont le but premier est d'assurer les télécommunications des utilisateurs).

L'interprétation des informations nécessite de posséder une capacité à les traiter simultanément en grande quantité. Aujourd'hui, les travaux dans le domaine de l'intelligence artificielle et ses divers exemples d'applications laissent augurer des solutions prometteuses pour la gestion de réseau (diagnostic, planification, etc). Ces solutions dépendent en grande partie des outils de gestion disponibles pour assister l'opérateur dans ces opérations.

1.4 Gestion des pannes

Il s'agit des fonctions que les normes recouvrent sous le terme « gestion des anomalies ». La détection de panne est indispensable pour que les mécanismes de réparation et de reconfiguration puissent se réaliser et laisser un système dans un état opérationnel. En outre, la détection à temps d'un dysfonctionnement et l'analyse rapide des phénomènes de défaillance permettent de prévenir l'aggravation de la situation et d'arrêter la propagation de ses conséquences par une intervention judicieuse et rapide. En général, la gestion des pannes suit les étapes suivantes :

1. *détection* : l'opérateur se rend compte d'un problème sur le réseau (apparition d'une panne) ;
2. *diagnostic* : il recherche la nature du problème (localisation et identification de la panne) ;
3. *réparation* : il commande une intervention si nécessaire.

1.4.1 Qu'est-ce qu'une panne ?

Dans les normes et dans la littérature scientifique, on trouve une grande diversité de termes pour signifier « les pannes en général », vues sous des angles différents : défaillance, incident, faute, dysfonctionnement, anomalie, etc. Dans la norme X. 733 [UT 92b] par exemple, on définit les termes suivants :

– *erreur* : une déviation du système par rapport à l'opération normale ;
– *faute* : une condition qui provoque un dysfonctionnement (et se manifeste par des erreurs).

Pour simplifier et pour bien comprendre le problème de la gestion des pannes dans les réseaux de télécommunications, [Boubour 97] propose une définition (reprise aussi dans [Aghasaryan 98]) de la panne adaptée à ce problème :

Définition 1.1 (Panne) *Une* panne *est un état de non fonctionnement ou de dysfonctionnement, matériel ou logiciel pertinent pour l'opérateur, au sens où il souhaite en avoir une trace dans le suivi.* □

Évidemment, cette définition se base sur la perception subjective de l'opérateur, et dépend du niveau de détail auquel il s'intéresse. Par exemple, certains problèmes transitoires peuvent être résolus par des mécanismes automatiques (utilisation des correcteurs d'erreurs etc), et naturellement ils ne seront pas considérés par l'opérateur comme des pannes à surveiller et à diagnostiquer.

Les pannes peuvent être classifiées comme *permanentes* ou *intermittentes* . Une fois produites, les pannes permanentes exigent une action de réparation. Par exemple, un câble sectionné entre deux équipements est une panne permanente. Les pannes intermittentes se manifestent de façon discontinue mais peuvent se répéter au cours du temps. Par exemple, la réinitialisation d'un équipement réseau est une panne intermittente. Elle peut se produire plusieurs fois au cours du fonctionnement du réseau. Le diagnostic des pannes intermittentes est une tâche plus complexe car les conséquences d'une panne de ce type peuvent disparaître.

On distingue aussi les pannes *primaires* des pannes *secondaires* [Hong et Sen 91]. Les premières constituent un ensemble de pannes indépendantes. Du point de vue du réseau ces pannes sont spontanées, par exemple, le fait qu'un câble soit sectionné est une panne primaire. Une telle panne peut être transmise aux autres entités par des liens matériels ou des liens logiciels. Les pannes secondaires apparaissent comme une conséquence d'une ou de plusieurs autres pannes (primaires ou secondaires). Les pannes primaires peuvent donc enchaîner une *propagation de pannes* causalement reliées. Par exemple, la panne primaire « section de câble » peut provoquer les pannes secondaires « réinitialisation d'équipement » sur les équipements

physiquement reliés par le câble [2].

En raison de la complexité et de l'hétérogénéité du réseau, ces relations de cause à effet ne sont pas toujours déterministes. Par exemple, une panne peut avoir des conséquences différentes en fonction du réglage de certains paramètres qui étant considérés comme des informations de très bas niveau ne sont pas pris en compte par le système de gestion. Ainsi, la propagation des pannes contient un degré d'incertitude vis-à-vis des connaissances de l'opérateur et des informations utilisées par le système de gestion. Une discussion intéressante sur la propagation de pannes et les aspects de non-déterminisme dans la gestion des pannes peut être trouvée dans les travaux [Wang 89] et [Hong et Sen 91].

1.4.2 Comment détecter une panne ?

L'origine d'une panne est détectée soit par un logiciel, soit par des mécanismes de capteurs, de « chiens de garde » internes ou encore par surveillance d'une unité par une autre. Cela est réalisé par des fonctions de surveillance et de prise en compte d'éléments non sollicités. Par exemple, la détection de pannes sur un élément du réseau peut se faire à l'aide d'un compteur qui vérifie que le taux d'erreurs de transmission sur cet élément ne dépasse pas un certain seuil. En général, des approches plus sophistiquées peuvent être employées dans le but de donner plus de précision sur la panne (et parfois même, l'identifier)[Wilsky 76][Bouloutas et al. 92]. Ces approches peuvent faire appel à des méthodes de traitement du signal [Basseville et Nikiforov 93]. Ces détecteurs permettent d'informer et d'attirer l'attention des gestionnaires de réseaux afin qu'ils puissent prévenir les dysfonctionnements. Ainsi, on peut détecter des pannes avant que leurs effets ne deviennent trop conséquents.

1.4.2.1 Notion d'alarme

L'information élémentaire pour la détection de panne est l'*alarme*. Dans [Arpège 92], la notion d'alarme est définie comme suit :

Définition 1.2 (alarme) *Une* alarme *est une indication de modification d'une condition qui peut avoir un impact négatif immédiat ou potentiel sur l'état de l'élément du réseau surveillé.* □

La norme X. 733 classifie les alarmes en plusieurs types et définit un champ « causes probables » de l'alarme (voir le tableau 1.1).

1.4.2.2 Signalisation d'alarmes

La norme X.733 définit la fonction de gestion de *rapport d'alarmes* [UT 92b] (on parle aussi de *signalisation d'alarmes*) qui permet de notifier qu'une alarme a été détectée. La détection d'une alarme se fait à un instant donné (remontée d'alarme). Une signalisation d'alarme est caractérisée par au moins quatre attributs :

[2] Parler de « panne primaire » ou de « panne secondaire » est un abus de langage, il s'agit de l'occurrence de la panne qui est primaire ou secondaire, et non la panne elle-même. Par exemple, la panne « réinitialisation d'équipement » peut être spontanée ou provoquée, certaines de ses occurrences seront donc primaires et d'autres secondaires.

Type d'alarmes	Causes probables (liste non exhaustive)
Communication	Perte de signal Erreur locale de transmission Erreur d'établissement d'appel
Qualité de service	Temps de réponse trop long Taux de retransmission trop important
Traitement logiciel	Erreur logicielle Place mémoire insuffisante
Équipement	Alimentation Interface Problème processeur
Environnement	Détection de fumée Température, ventilation Feu

TAB. 1.1 – Types d'alarme et leurs causes probables (norme UIT-T X. 733).

1. le type de l'alarme ;
2. l'identificateur du composant émetteur ;
3. la cause probable ;
4. la date.

Le premier attribut indique le type d'alarmes (voir le tableau 1.1). Le second attribut indique l'endroit du réseau (selon le standard d'adressage accepté) et la ressource où la panne a été détectée. La cause probable sert à la description du dysfonctionnement découvert. C'est une traduction en texte du code d'erreur, et malgré son titre souvent elle n'indique pas vraiment la cause du problème. Par exemple, la cause probable d'une alarme de communication générée sur une entité x indique la perte de signal provenant de l'entité y, sans rien dire sur le pourquoi de cette perte. Le troisième attribut est la date d'occurrence de la détection. Les composants d'un domaine de gestion peuvent être synchronisés par une horloge locale, mais en général, pour les réseaux qui couvrent de grands espaces géographiques, il n'existe pas forcément de moyens de synchronisation avec une horloge globale : cet attribut de date n'est donc pas forcément un moyen de comparaison avec les dates d'occurrences des autres alarmes non locales.

1.4.2.3 Exemple d'architecture de signalisation

La gestion des alarmes au niveau d'un élément du réseau consiste à analyser les anomalies (filtrage, corrélation, etc.) au niveau local puis à émettre au superviseur les informations les plus pertinentes et les plus expressives. L'intelligence d'un élément de réseau tient en sa capacité à ne pas générer des alarmes parasites et à son pouvoir d'extraire des informations significatives à partir d'un ensemble d'alarmes. Le schéma de la figure 1.5 décrit l'architecture fonctionnelle du processus d'alarmes extrait de la norme X.734 [UT 92c]. Ce processus passe par plusieurs phases. La figure 1.5 donne un aperçu des tâches non exhaustives qui doivent être prises en compte.

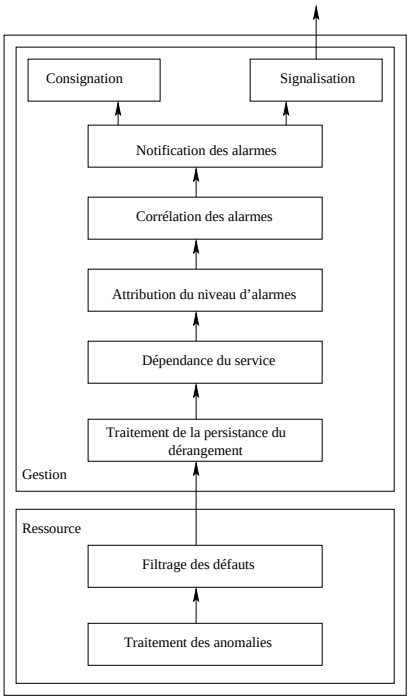

FIG. 1.5 – Exemple d'architecture fonctionnelle du processus d'alarmes extrait de la norme X.734

Un filtre temporel est placé à l'entrée de la partie gestion. Il permet d'attendre un certain temps avant de générer les alarmes. Seul le dysfonctionnement qui passe ce filtre est signalé comme alarme. Un niveau de priorité est attribué à chaque alarme, ce qui permet de déterminer le type de notification à émettre (exemple : une alarme mineure pour un événement n'affectant pas le service et une alarme critique pour un événement affectant l'état du service). Le niveau corrélation d'alarmes permet de ne signaler que les causes premières des événements. L'étude des corrélations d'alarmes est une phase très importante dans le processus de gestion d'alarmes. L'intelligence d'un élément du réseau dépend essentiellement des techniques et de l'efficacité de synthèse des alarmes. Après les calculs de corrélation, les alarmes sont filtrées en fonction de leurs contenus, tels que le type et la cause de l'alarme, le niveau, etc. Les alarmes filtrées sont émises vers une ou plusieurs destinations (superviseur).

1.4.3 Comment diagnostiquer une panne ?

En général, le contenu d'une alarme ne suffit pas à identifier la panne ni à décider des actions de réparation à entreprendre.

D'une part, cela s'explique par le fait que la panne ne se trouve pas forcément dans le composant qui la détecte (ou plutôt qui détecte ses conséquences). Les processus de surveillance qui effectuent la détection de pannes ne disposent que d'informations concernant un composant (ou un groupe local de composants), or plusieurs pannes extérieures peuvent se manifester par

des symptômes identiques sur un élément du réseau donné.

D'autre part, même si physiquement la panne se trouve dans le composant où la détection se fait, son identification peut être impossible par manque de détails concernant les informations disponibles. Dans ce cas, les symptômes observés sur des composants voisins peuvent être décisifs pour déterminer la source du problème.

Ainsi, le plus souvent, les alarmes générées par les processus de détection n'identifient pas les pannes et ne déterminent pas exactement leur localisation ; d'où la nécessité du diagnostic qui prend en compte l'ensemble des informations disponibles dans le réseau de gestion.

Le processus de diagnostic peut consulter des événements du passé (des alarmes ou d'autres notifications) pour en déduire des propagations de pannes sous-jacentes. Le nombre des alarmes générées et l'ambiguïté des informations qu'elles portent augmentent de façon cruciale avec la taille et la complexité des réseaux de télécommunications. Le diagnostic des pannes est donc une tâche difficile et devient un défi très important. Dans le cas où les résultats du diagnostic ne sont pas suffisants pour la localisation et l'identification des pannes, pour les préciser ou de les confirmer, des tests peuvent être déclenchés. Les composants susceptibles d'être en panne selon les résultats du diagnostic sont alors testés par cette procédure dite « test de diagnostic ».

1.4.4 Comment réparer une panne ?

En fonction de la nature de la panne, la réparation est effectuée soit par intervention d'opérateurs humains, soit par le système de gestion.

L'intervention humaine a principalement lieu lorsqu'il s'agit d'une panne physique (équipement, câble, etc). Une intervention humaine donne lieu à un dossier d'intervention. Il permet de suivre l'avancement du traitement du problème, et surtout *a posteriori*, d'analyser les résultats, de déterminer si des fausses manœuvres ont eu lieu ou si la procédure de traitement a été satisfaisante. Ce dossier peut contenir des informations du type : problème, alarmes associées, essais et mesures effectués, actions entreprises, opérateurs en cause et autres dossiers éventuellement corrélés.

Il se peut aussi que la réparation soit assurée par un système automatique. Par exemple, dans le cadre de Transpac, des stations exécutent des processus réseaux (rôles) assurant la transmission des données. Si l'une de ces stations tombe en panne, le système de gestion bascule sur une station de secours pour assurer le même service.

1.4.5 Gérer les pannes, c'est superviser le réseau

La simple signalisation d'alarmes provenant des éléments gérés ne suffit pas pour en déduire directement un diagnostic des pannes sur le réseau entier (cf section 1.4.3). Il est nécessaire d'avoir une connaissance plus globale du réseau. Aussi, la mise en place d'un diagnostic repose sur les services proposés par la supervision du réseau (surveillance continue des alarmes), voir figure 1.6.

1.4.5.1 Notion de supervision

Le réseau possède son propre système de défense intégré, chargé de détecter et de pallier certaines pannes. La supervision s'effectue donc sur l'ensemble composé du réseau et de son système de défense. Superviser c'est être capable, à partir des alarmes :

- d'être au courant de l'état de fonctionnement de chacun des éléments du réseau ;
- d'observer l'évolution du réseau au cours du temps, d'indiquer à l'opérateur les phénomènes les plus marquants de cette évolution, de vérifier qu'elle est conforme à ce que l'on attend du réseau en fonctionnement, et si ce n'est pas le cas, d'en avertir l'opérateur.

Le raisonnement de supervision s'effectue essentiellement sur les alarmes. Cependant, il est aussi possible d'interroger le réseau de manière active pour obtenir l'état d'un composant. Ces interrogations ont un coup sur les performances du réseau supervisé. Il est donc souhaitable de les minimiser.

1.4.5.2 Les difficultés de la supervision

Compte tenu de l'existence du système de défense intégré et du nombre d'émissions spontanées, on peut se demander où se trouve le problème de supervision. En effet, lorsqu'un composant tombe en panne, il émet une alarme et lorsqu'il repasse en état de fonctionnement, il réémet une alarme : suivre son évolution semble donc relativement simple. Néanmoins, la supervision est sujette à certaines difficultés.

- Les *alarmes parasites*. Les alarmes significatives sont « noyées » dans un flot d'alarmes qui ne traduisent qu'un fonctionnnement normal du système. La première difficulté de la supervision est donc de retrouver parmi ce flot les alarmes qui sont la conséquence de dysfonctionnements.
- Le *masquage d'alarme*. Le phénomène de masquage est la difficulté principale du problème de supervision. Lorsqu'un composant tombe en panne ou revient en état de fonctionnement, il émet une alarme. Mais ceci ne signifie pas qu'elle arrivera nécessairement au superviseur. En effet, pour arriver au superviseur, une alarme doit transiter par un certain nombre de composants. Il est possible que l'un des composants par lequel doit transiter l'alarme ne soit pas en état pour la retransmission. L'alarme est dite masquée. Ce phénomène de masquage rend incomplet l'ensemble des alarmes reçues par le superviseur.
- La *perte d'alarme*. Si plusieurs alarmes arrivent en même temps sur un même composant, elles sont stockées dans des tampons. La taille de ceux-ci est limitée et au-dessus d'un certain seuil les alarmes arrivant sont perdues.

Ces phénomènes de masquage et de perte peuvent se produire pour plusieurs raisons, liées à l'occurrence de certaines pannes.

- Les *pannes multiples*. On parle de pannes multiples lorsque plusieurs pannes ont des

effets qui se chevauchent dans le temps. Une occurrence de panne est associée à un début et à une fin d'occurrence. On dit qu'il y a phénomène de pannes multiples lorsque les intervalles de début et de fin de pannes se chevauchent dans le temps. Le fait que des pannes peuvent se produire en même temps peut conduire à un phénomène de masquage. Par exemple, si une panne de rupture de lien a lieu, si une deuxième panne apparaît devant produire une alarme sur le lien rompu, cette alarme sera masquée à cause de la première panne.

– Les *pannes corrélées*. Les pannes corrélées ont lieu à cause du phénomène de propagation des pannes. Les pannes corrélées peuvent produire des cascades d'alarmes pouvant être sujettes à des pertes.

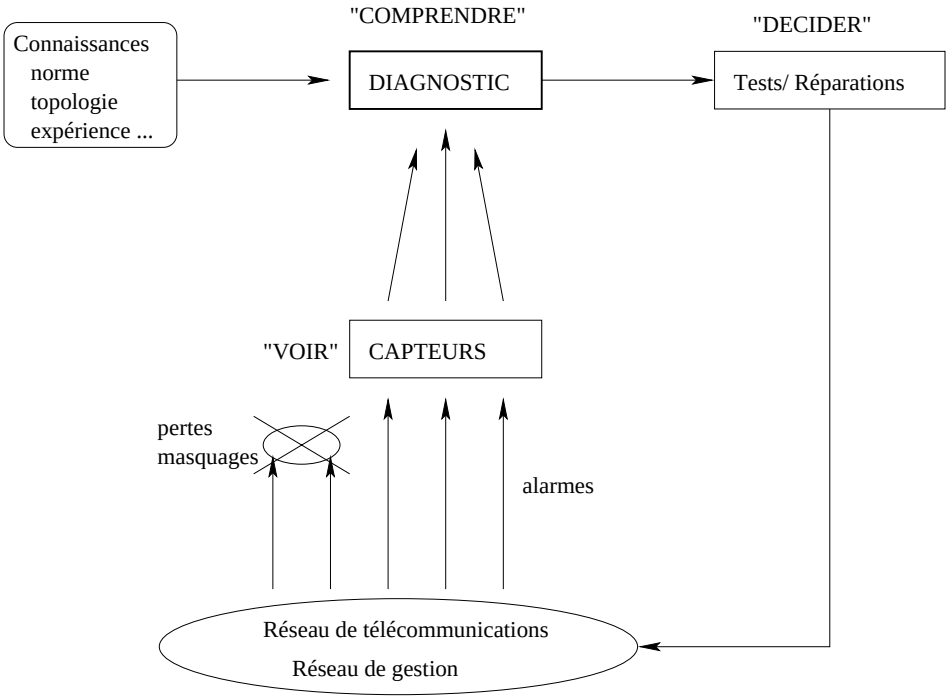

FIG. 1.6 – Cycle de la supervision et le diagnostic de réseau de télécommunications.

1.4.6 Conclusion

Le fonctionnement des réseaux de télécommunications est généralement très fiable. Ceci est dû à l'utilisation de mécanismes de protection performants au niveau des couches basses (codes détecteurs et correcteurs d'erreurs, contrôle de congestion, mécanismes de reprise, reroutage, etc.). De plus, la plupart des composants matériels sont fiables et associés à du matériel de secours. Par exemple, la duplication de stations dans le réseau Transpac joue ce rôle de sécurité. Cependant, tous les problèmes ne peuvent pas être résolus au niveau des couches basses. C'est par exemple le cas des pannes multiples et des pannes corrélées. Le diagnostic de ces pannes nécessite une analyse approfondie des comportements par la supervision du

réseau. Les phénomènes de masquage, de perte d'alarmes et la sophistication des équipements qui leur permet d'envoyer des messages de plus en plus nombreux, sont des difficultés que les opérateurs ne peuvent surmonter seuls. Il est donc nécessaire d'utiliser des techniques automatiques pour permettre une meilleure exploration des alarmes ainsi qu'une meilleure interprétation. Cette interprétation doit pouvoir se faire parfois dans l'urgence afin d'éviter une trop grande dégradation des services pour les utilisateurs du réseau.

CHAPITRE *2*

Diagnostic : les approches existantes

2.1 Introduction

Dès l'apparition des réseaux de télécommunications, leur fiabilité est devenue un problème crucial. Le diagnostic de pannes est alors devenu un sujet de recherche important qui intéresse de plus les industriels. En effet, des outils de diagnostic permettent de détecter plus rapidement les éventuels problèmes voire de les anticiper, ce qui a pour conséquence de minimiser le coût de la maintenance et de la réparation et d'augmenter la fiabilité du système.

Les travaux de recherche sur le diagnostic de pannes dans les systèmes dynamiques sont très nombreux. Les approches étudiées sont très diverses et chacune est révélatrice de son époque. Ces approches dépendent essentiellement des technologies et des ressources informatiques disponibles au moment de leur développement. Ces technologies se diversifiant et les ressources informatiques augmentant, les approches développées au cours du temps utilisent des informations de plus en plus granulaires pour obtenir des diagnostics de plus en plus riches.

Dans ce chapitre, nous présentons brièvement les différentes approches de diagnostic pour les systèmes dynamiques, qui ont été étudiées au cours de ces dernières décennies afin d'en dégager les avantages et les inconvénients.

2.2 Systèmes experts

La technique la plus répandue pour la supervision de réseaux est l'utilisation de systèmes experts [Sloman 94]. Les systèmes experts traditionnels à base de règles se présentent sous forme d'associations empiriques entre effets et causes représentées par des règles. Ces associations sont généralement fondées sur l'expérience de l'expert plutôt que sur une connaissance de la structure et du comportement du système (les systèmes experts font partie des systèmes dits *à connaissance de surface*).

La fonctionnalité d'un système expert dans la supervision d'un système est de trouver la cause de ce qui a été observé en parcourant les règles par des techniques classiques en IA telles que le chaînage avant, le chaînage arrière ou encore le chaînage mixte.

Dans la section suivante, nous présentons un système expert développé pour la supervision d'un réseau français de télécommunications à commutation de paquets : le réseau Transpac. Cet exemple va nous permettre de montrer quels sont les avantages et les inconvénients d'une telle approche.

2.2.1 Système expert du réseau Transpac

Le système d'aide à la supervision utilisé sur le réseau Transpac était à la base un système expert comportant environ 200 règles. Il effectue une synthèse des éléments provenant du réseau, propose des actions à l'opérateur, attire son attention sur des pannes trop fréquentes ou trop longues et met à jour une base de données des états des éléments du réseau. Cette base de données est mise à jour sans effectuer de photographies d'état, c'est-à-dire sans aller demander à chaque composant dans quel état il se trouve.

Ce système expert utilise des règles de production mises en œuvre à l'aide d'un générateur de systèmes experts : Chronos. Ce générateur est un outil de développement de systèmes experts. Il a été choisi pour résoudre le problème d'efficacité et pour faciliter l'intégration de l'expertise. Une règle type est constituée d'une partie prémisse et d'une partie conclusion. La partie prémisse décrit une suite d'alarmes, précisant pour chacune d'entre elles leur nature, leur provenance, et éventuellement leur date de réception et des délais entre ces dates, ou encore l'indication d'un nombre minimum d'alarmes. La partie conclusion indique en général les éléments indésirables survenus dans le réseau et supposés responsables de l'émission des alarmes. Elle peut aussi mentionner des actions à entreprendre par le superviseur (envois de commandes). Dans l'exemple ci-dessous, nous présentons en langage naturel, une règle typique décrite dans le système expert de Transpac. Dans cette règle entrent en jeu un composant du réseau de type CT (Centre Technique), et des commutateurs (un centre technique gère un ensemble de commutateurs).

Exemple [Une règle du système expert]

Dès que :
On a reçu une alarme CVHS concernant un objet $< x >$ du réseau du type CT au temps $T1$

et

On a reçu une alarme CVES concernant le même objet $< x >$ au temps $T2$ avec $T2 > T1$

et

Durant la période $[T1, T2 + 30 secondes]$, on a reçu plus de 3 alarmes de type N004 concernant des commutateurs dépendant de cet objet $< x >$

Faire :
Afficher à l'intention de l'opérateur « Il y a eu arrêt du $CT < x >$ du temps $T1$ au temps $T2$ »

La prémisse de cette règle exprime la réception d'un certain nombre d'alarmes (CVHS, CVES et N004) dans un intervalle de temps précis, provenant d'un centre technique $< x >$ et d'un sous-ensemble de ses commutateurs. La conclusion de cette règle est un diagnostic de panne envoyé à l'opérateur. Si une commande à activer existait dans ce cas de panne, la conclusion de cette règle serait augmentée d'un message informant l'opérateur que cette commande pourrait être activée pour résoudre le problème.

2.2.2 Avantages des systèmes experts

Une règle dans un système expert spécifie une partie du raisonnement que doit avoir l'opérateur de supervision. La qualité première d'un système fonctionnant avec de telles règles est son efficacité au niveau temps de calcul. Il suffit à un tel système d'attendre que survienne une succession d'événements extérieurs facilement observables puis de « sauter » directement aux conclusions [Ungauer 93]. Il n'a aucun raisonnement compliqué et coûteux en temps de calcul à tenir, aucun calcul intermédiaire à effectuer. Ceci est possible car il y a eu un expert, l'expert qui a produit cette règle qui, une fois pour toutes, a tenu ces raisonnements. Cet expert n'a ensuite enregistré dans le système que les conditions initiales et les conclusions finales de son raisonnement. On peut donc voir ses règles comme des raccourcis efficaces de raisonnements généralement beaucoup plus longs.

Puisque ces règles sont le produit d'experts humains, le résultat est aussi compréhensible pour l'opérateur. Ainsi, une règle d'un système expert est directement interprétable par l'opérateur et peut lui servir d'explication, de justification face à la situation à laquelle il est confronté.

L'implantation d'un système expert est aussi très simple. En effet, des outils de génération de systèmes experts (tels que Chronos) ont été développés et facilitent le travail (pas d'algorithmes à développer, il suffit juste de rentrer les règles dans le langage reconnu par le générateur utilisé).

Tous ces atouts ont fait que les industriels se sont intéressés à la mise en place de systèmes experts pour la supervision de leur procédés industriels. En France, par exemple, le système Sachem met en œuvre une telle approche : ce système est utilisé actuellement pour surveiller et contrôler des hauts-fourneaux ainsi qu'une ligne de galvanisation [Frydman et al. 01]. On peut aussi citer le projet Alexip de l'IFP (Institut Français du Pétrole) : ce projet propose un environnement générique à base de connaissances pour la supervision de procédés de raffinage et de pétrochimie [Cauvin et al. 92]. Dans la gestion des réseaux, outre le système de Transpac, divers systèmes ont été développés à travers le monde. On peut citer par exemple le système Noaa (*Network Operations Analyzer and Assistant*) : il s'agit d'un système expert pour la gestion du trafic dans le réseau téléphonique de *Pacific Bell* [Goodman et al. 95].

2.2.3 Inconvénients des systèmes experts

Ce qui fait la force d'un système expert, c'est le jeu de règles efficaces résultat de l'expertise d'un humain. Mais c'est aussi le premier inconvénient d'une telle approche : elle est totalement dépendante de l'expertise faite sur le système à superviser. Ainsi, les systèmes experts sont sujets aux défauts liés à l'expertise elle-même.

– *La difficulté d'acquisition de l'expertise*. Ce point est particulièrement important lors de l'installation d'un nouveau réseau. En effet, puisque le réseau est nouveau, il n'y a pas ou peu d'expériences au sujet des pannes pouvant se produire et surtout des événements (observables en particulier) qui peuvent en être les conséquences. Pour une bonne expertise d'un système, il faut du temps si bien qu'un système expert ne peut être opérationnel dès le début de l'exploitation d'un réseau.

- *Le manque de généricité*. Les règles acquises sur un réseau ne peuvent être utilisées sur un autre réseau car elles sont trop souvent dépendantes de l'architecture du réseau.
- *Le problème de l'évolution du système*. Si le système à superviser évolue (ce qui est souvent le cas dans les réseaux de télécommunications) soit par remplacement de composants, soit par des ajouts de composants, le système de règles est à remettre en cause. Une nouvelle expertise doit être mise en place afin que le système expert soit toujours pertinent face aux observations.

Aux inconvénients liés à l'expertise s'ajoutent des problèmes liés à l'approche même [Gurer et al. 95].

- *Robustesse*. Les règles sont fixées et ne sont pas robustes face à des situations non reconnues.
- *Données incertaines*. Les systèmes experts ne sont pas intéressants s'il faut manipuler des probabilités et de l'incertitude. Ils ont des difficultés dans l'analyse d'un ensemble important de données non corrélées, ambiguës et incomplètes. Le domaine des règles doit être bien compris et pensé. Ceci n'est pas forcément possible dans des domaines tels que la gestion de panne.
- *Manque de connaissances profondes*. La seule information que retourne un système expert est la conséquence d'une règle reconnue. Il ne donne pas en général une explication des conclusions adoptées (par exemple, une information sur la propagation des pannes dans le système).
- *Incohérence des règles*. L'ajout ou la suppression d'une règle peut avoir un impact sur d'autres règles, impact qui est difficile à détecter.

2.2.4 Conclusion

L'approche des systèmes experts est un succès dans le domaine de la supervision. Beaucoup d'industriels ont déjà développé de telles techniques. Une telle approche est intéressante de par son efficacité et sa facilité de développement dans un monde industriel. Dans le domaine des télécommunications, le diagnostic de panne à l'aide d'un système expert a eu son heure de gloire. Néanmoins, face à l'essor des télécommunications et donc à l'évolution quasi-permanente des réseaux de télécommunications, une telle approche n'est plus appropriée : le temps nécessaire à l'expertise pour élaborer de nouvelles règles n'est pas négligeable par rapport au temps entre deux évolutions (par exemple, l'expertise de Transpaca demandé deux années). L'évolution des règles d'un système expert est trop coûteuse (temps d'expertise, temps durant lequel le système expert n'est pas exploitable) pour qu'une telle approche soit viable désormais pour la gestion des pannes dans un réseau.

Certains travaux ont été menés afin de résoudre le problème de la trop grande dépendance entre les règles, le système et son expertise par un opérateur humain. Leur but est d'acquérir des règles en se passant plus ou moins de l'expert (l'objectif de l'expert dans ce contexte n'est plus de découvrir des règles mais plutôt de valider les règles acquises automatiquement). L'acquisition de ces règles passent en général par un apprentissage qui se fonde sur un *modèle* du système à superviser (le cœur humain [Bratko et al. 88], le système d'alimentation électrique d'un satellite [Pearce 88], ou bien encore un réseau de transmission de données radio GPS [Smyth et al. 91]). Une autre approche consiste à raisonner en se fondant sur des cas de pannes

déjà expertisés [Lewis 93]. Lors de l'analyse d'un nouveau cas de panne, on recherche les antécédents similaires pour proposer des solutions à un expert ; celui-ci les analyse et retourne une solution, cette solution est incorporée dans la base des cas, ce qui augmente la robustesse du système de diagnostic.

Une tendance actuelle est de considérer le système expert non plus comme un système unique pour la supervision mais comme un outil combiné avec d'autres. Par exemple, dans [Gurer et al. 95], le système expert est utilisé comme un système de filtrage d'alarmes (redondances, etc). Dans ce contexte, l'information demandée à l'expert est de plus bas niveau, plus localisée et donc plus stable face à l'évolution du système à superviser. Pour traiter une information de plus haut niveau (propagation de pannes dépendant de la topologie par exemple), il est alors nécessaire d'utiliser des techniques plus générales : la *corrélation d'alarmes* ou le *diagnostic à base de modèles*.

2.3 Corrélation d'alarmes

Dans la littérature sur la gestion des pannes dans un réseau de télécommunications, les méthodes de diagnostic de pannes apparaissent souvent sous le nom de méthode de *corrélation d'alarmes*. En fait, la notion de corrélation d'alarmes est plus générale mais l'une des tâches principales qu'elle est en mesure de résoudre est le diagnostic de pannes proprement dit.

2.3.1 Notions sur la corrélation d'alarmes

Dans le domaine de la gestion des pannes dans les réseaux de télécommunications, la corrélation d'alarmes est une vision très naturelle. Par ailleurs, la meilleure façon de présenter la corrélation dans ce cadre est d'en donner un exemple (très largement inspiré de [Nygate 95]). Le réseau que l'on considère est un ensemble de commutateurs. On considère en particulier trois de ces commutateurs. Le premier est relié à chacun des deux autres par deux connexions (cf figure 2.1).

La chronologie des pannes est la suivante. A 4h24, une panne physique P1 se produit sur la connexion L4, à 4h50 a lieu une requête de trafic important sur le commutateur 2 qui s'est bloqué (P2) et à 5h22 une deuxième panne physique P3 a lieu sur la connexion L3. Dans cet exemple, les alarmes générées liées aux pannes P1, P2 et P3 sont au nombre de 19 (cf tableau 2.1). Parmi ces 19 alarmes, il y en a une, en particulier, qui informe que le commutateur 1 est isolé. Ces 19 alarmes ont un lien commun, la présence de P1, P2 et P3, elles sont dites *corrélées*. Comme le réseau est un système important, de nombreux événements indépendants de P1, P2 et P3 peuvent avoir lieu pendant cette période, si bien que les 19 alarmes en question peuvent être « noyées » dans une liste beaucoup plus importante d'alarmes reçues par le superviseur. L'objectif d'un système de corrélation est donc de renseigner l'opérateur sur la présence des 19 alarmes qui dénotent, par *corrélation*, l'occurrence des pannes P1, P2 et P3 et qui expliquent ainsi pourquoi le système a notifié que le commutateur 1 est isolé.

On appelle *corrélation d'alarmes* (et plus généralement *corrélation d'événements*) la tâche qui consiste à interpréter conceptuellement un ensemble d'alarmes (plus généralement, un ensemble d'événements) afin d'en extirper une signification, une information plus riche

Date	Composant	Alarme
4:24	Liaison L4	Problème physique
4:27	Commutateur 1	Rupture liaison L4
4:29	Commutateur 3	Rupture liaison L4
4:50	Commutateur 2	Requête Trafic important
4:52	Commutateur 2	Réinitialisation L1
4:54	Commutateur 1	Surcharge L1
4:55	Commutateur 1	Surcharge L2
4:56	Commutateur 2	Réinitialisation L2
4:56	Commutateur 2	Rupture liaison L1
4:58	Commutateur 2	Rupture liaison L2
5:00	Commutateur 2	Connexion 5 perdue
5:02	Commutateur 1	Rupture liaison L1
5:04	Commutateur 1	Rupture liaison L2
5:06	Commutateur 2	Connexion 31 perdue
5:22	Liaison L3	Problème physique
5:24	Commutateur 1	Rupture liaison L3
5:26	Commutateur 1	Connexion 1 perdue
5:28	Commutateur 1	Commutateur isolé
5:30	Commutateur 3	Rupture liaison L3

TAB. 2.1 – Alarmes générées par le réseau de la figure 2.1.

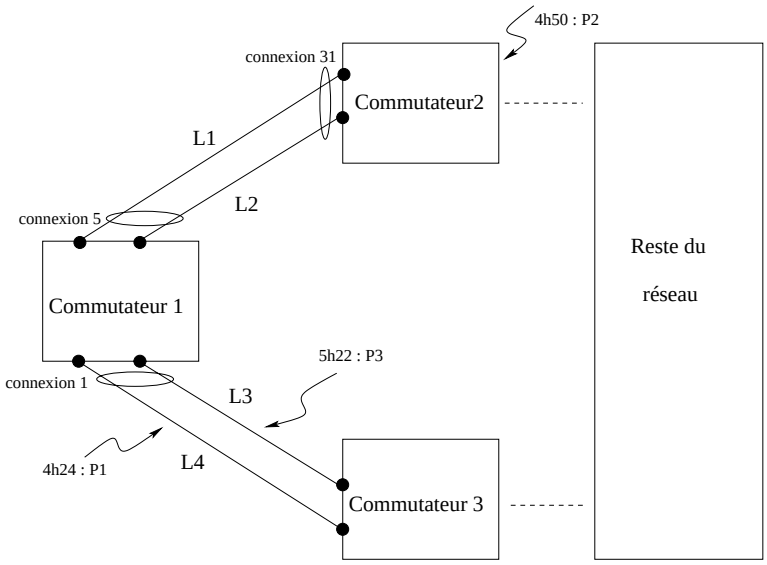

FIG. 2.1 – Exemple de réseau et d'occurrence de pannes.

[Jakobson et Weissman 93]. Généralement, dans un système de corrélation, on définit un ensemble de *règles de corrélation* qui sont des associations entre un ensemble d'*événements corrélés* et l'information ou la signification qu'on veut apporter lorsqu'un tel ensemble d'événements a été *reconnu*. On dit aussi qu'un tel ensemble d'événements reconnus définit une *corrélation* (cf figure 2.2).

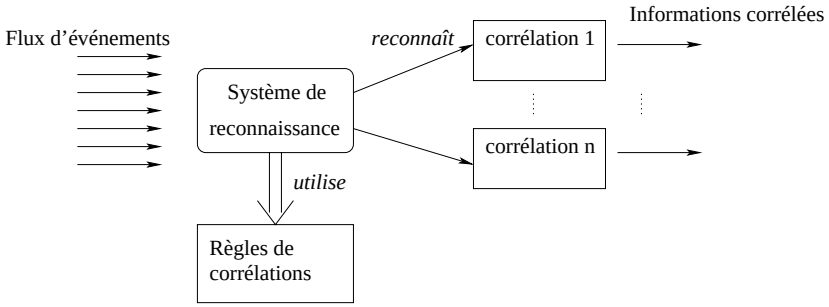

FIG. 2.2 – Principe de la corrélation d'événements.

La corrélation est un processus générique qui peut servir à accomplir plusieurs tâches dans la gestion de réseaux.

– Le *filtrage* : cette tâche consiste par exemple à réduire plusieurs occurrences d'une même alarme en une seule (*compression*), à inhiber certaines alarmes (*suppression*), à remplacer un motif reconnu d'alarmes par une alarme unique (*substitution*) ou à remplacer une alarme d'un certain type par une alarme d'un type plus général, autrement dit avec moins de paramètres (*généralisation*).

– La *localisation* et l'*identification* de pannes.
– La sélection d'actions correctrices.

2.3.2 Architectures des systèmes de corrélations

En toute évidence, la technique des systèmes experts utilisée dans des tâches de supervision (voir section 2.2) fait partie des méthodes fondées sur la corrélation d'alarmes. Néanmoins, il existe des systèmes de corrélation propres.

2.3.2.1 ECXpert

Le système ECXpert (*Event Correlation Expert*) a été réalisé spécifiquement pour effectuer de la corrélation d'alarmes [Nygate 95] dans un réseau de télécommunications. De nombreux dysfonctionnements d'un réseau peuvent être caractérisés par des séquences typiques. Les différentes alarmes d'une séquence possèdent alors des relations de causes à effets. Dans ECXpert, de telles séquences sont représentées par un *squelette d'arbre de corrélations* (*correlation tree skeleton*, voir figure 2.3).

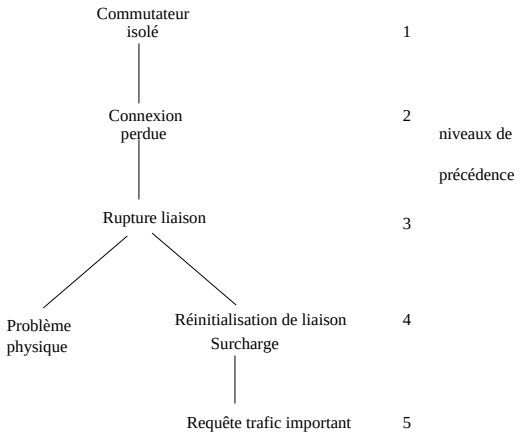

FIG. 2.3 – Squelette d'arbre de corrélations.

Dans ces arbres, les liens enfant/parent sont équivalents à des relations de cause à effet entre alarmes, par exemple la réception de l'alarme *Connexion perdue* peut avoir pour conséquence la réception de l'alarme *Commutateur isolé* (mais ce n'est pas forcément toujours le cas). Des alarmes équivalentes (telles que *Réinitialisation* et *Surcharge*) sont sur le même nœud de l'arbre. Si un nœud dispose de plusieurs fils, cela signifie que chaque fils peut être indépendamment la cause du nœud parent. Par exemple, la réception de l'alarme *Rupture liaison* peut être une conséquence de l'apparition d'une alarme du type *Problème physique* ou *Surcharge*.

À partir de cette structure d'arbre, ECXpert établit des *instances d'arbres de corrélations* en fonction des alarmes reçues. La figure 2.4 présente une telle instance lorsque le système reçoit un flot d'alarmes contenant les alarmes présentées dans le tableau 2.1.

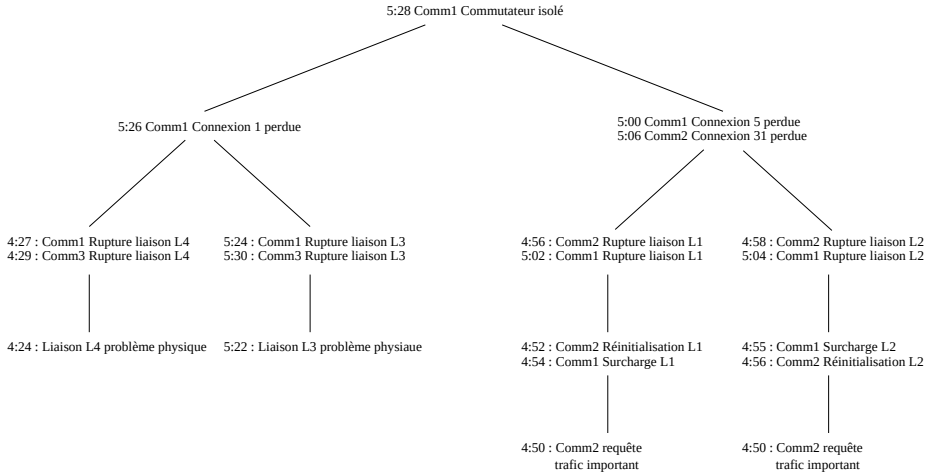

FIG. 2.4 – Une instance d'arbre de corrélation résultat de ECXpert.

Chaque nœud contient un groupe d'alarmes équivalentes et deux nœuds sont connectés s'il y a une relation de cause à effet entre eux ; autrement dit, chaque branche de l'instance correspond à une branche du squelette (figure 2.3). Chaque feuille est donc considérée comme une *alarme primaire*, conséquence directe de l'occurrence d'une panne primaire (*Problème physique* sur L4 correspond à l'occurrence de la panne P1, *Requête Trafic important* sur le commutateur 2 à celle de P2 et *Problème physique* sur L3 à celle de P3).

Le système ECXpert est écrit en C++/Prolog. La partie C++ permet de gérer les différentes structures de corrélation (les arbres), quant à Prolog, il est utilisé pour produire l'association entre les alarmes reçues et les règles de corrélations disponible grâce à ses capacités algorithmiques de chaînage arrière (technique des systèmes experts classique). Il permet de gérer 1000 alarmes par heure en utilisant dix groupes de corrélations.

2.3.2.2 Impact

Le système Impact (*Intelligent Managemenent Platform for Alarm Correlation Tasks*) est un système de corrélation similaire à EXCpert [Jakobson et Weissman 93]. L'objectif de ce système de corrélation est triple :

1. le filtrage d'alarmes ;
2. la généralisation d'alarmes ;
3. le diagnostic de pannes.

Le point important dans ce système est qu'il présente une vision hiérarchique de la corrélation d'alarmes (cf figure 2.5).

Les corrélations sont décrites à l'aide de classes (une corrélation est représentée par une instance de classe). Chaque corrélation décrit l'état du réseau en se fondant sur l'interprétation des événements du réseau (les alarmes, qui sont elles-mêmes représentées par des instances de classes). Non seulement, une corrélation contient des événements du réseau mais aussi d'autres

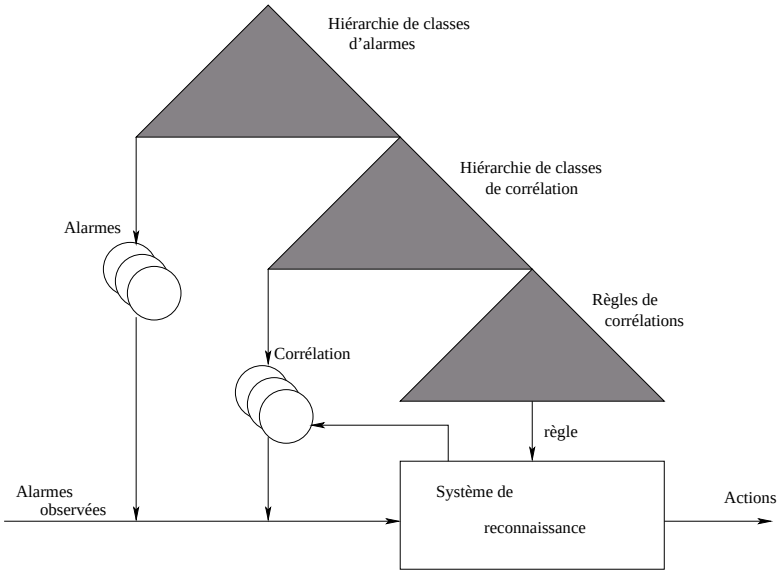

FIG. 2.5 – Hiérarchie conceptuelle de la corrélation d'alarmes.

corrélations, si bien que la description d'une corrélation peut être plus abstraite car décrite par rapport à d'autres corrélations. L'avantage de cette vision hiérarchique vient du fait même que les réseaux de télécommunications sont des systèmes très hiérarchiques, il est donc plus aisé de décrire des règles de corrélations avec cette vision. En reprenant l'exemple décrit sur les figures 2.1 et 2.3, on peut se rendre compte de la hiérarchie. Par exemple, les alarmes *Rupture liaison* et *Connexion perdue* ne se situe pas au même niveau. L'alarme *Rupture liaison* est associée à un problème physique alors qu'une alarme *Connexion perdue* est associée à un problème protocolaire (routage, etc).

2.3.2.3 Mise en œuvre efficace d'un système de corrélation

[Kliger et al. 95] part du constat que les systèmes de corrélations classiques (tels que ceux présentés précédemment) sont sujets à plusieurs problèmes, en particulier l'efficacité ainsi que la robustesse face au bruit pouvant être contenu dans les observations. Ces problèmes sont liés au fait que les moteurs de corrélations traitent directement les événements afin de détecter les corrélations. [Kliger et al. 95] propose alors une mise en œuvre du système de corrélation qui répond à ces critères d'exigence. L'idée consiste à précompiler les corrélations sous forme d'un code (un vecteur), ce code représentant un ensemble minimal d'observations, de symptômes, permettant d'établir une corrélation unique (cet ensemble est appelé *codebook*). De même, lors de la réception des alarmes, elles sont codées sous forme de vecteurs dans le même domaine de représentation que le code des corrélations. La reconnaissance d'une corrélation consiste alors à confronter le code observé à l'ensemble des codes de corrélations, les corrélations retenues étant celles dont l'association avec le vecteur d'observations maximise une *mesure de corrélation* entre les codes.

2.3.3 Approches à base de reconnaissance de forme

Il existe d'autres approches de corrélations d'alarmes qui se fondent sur les techniques de reconnaissance de forme. La reconnaissance des formes a pour but en effet la reconnaissance d'une forme parmi différentes possibilités à partir d'observations bruitées de celles-ci [sldd 01]. Si on applique les termes de la corrélation d'alarmes à la reconnaissance de formes, alors l'information associée à une corrélation (à savoir le diagnostic) est une « forme », et les observations de cette forme sont les alarmes.

2.3.3.1 Principe

Le problème du diagnostic à base de reconnaissance de forme se pose ainsi :
- on définit un vecteur forme **x** représentatif de l'état du système à diagnostiquer ; ce vecteur est constitué de paramètres observables du système ;
- on définit un ensemble de pannes possibles du système, chaque panne constitue une classe de vecteurs formes (une zone de l'espace des vecteurs formes) ;
- on construit une règle de décision $d(\mathbf{x})$ qui, au vecteur **x**, associe la décision d'affecter **x** à une des classes ou non (règles de corrélation).

Contrairement aux autres domaines où la reconnaissance de formes est utilisée (reconnaissance de caractères, d'images...) et où le nombre de classes est connu *a priori*, dans le domaine du diagnostic, ce n'est pas le cas. En effet, dans des problèmes réels, le nombre d'états de pannes (c-à-d de classes) est très important, de plus avec les observations dont on dispose, il n'est pas forcément possible de discriminer les différentes classes. Aussi, dans le cadre du diagnostic à base de reconnaissance de formes, le système de décision doit pouvoir admettre le *rejet* :
- soit un *rejet de distance*, à savoir que le vecteur forme à reconnaître est trop éloigné des classes connues, il faut donc décider de l'affecter à une nouvelle classe inconnue ;
- soit un *rejet d'ambiguïté*, à savoir que le vecteur forme peut appartenir à deux classes distinctes avec des probabilités ou certitudes similaires, il ne faut donc pas décider de l'affecter à l'une ou à l'autre.

2.3.3.2 Diagnostic de perturbations dans un réseau de télécommunications

Dans [Didelet 92], [Didelet et Dubuisson 92], les auteurs utilisent une telle approche pour diagnostiquer des perturbations dans le réseau téléphonique commuté (RTC) de *France Telecom*. Ils définissent des classes de pannes telles que : *situation nominale*, *surcharge globale du réseau*, *rupture de faisceaux*... Les composantes des vecteurs formes sont établies à partir de certaines observations du réseau ; par exemple les *prises efficaces* (nombre d'appels présentés dans un centre qui ont été suivis d'une sonnerie chez le destinataire) ou bien encore le *taux d'occupation* (fonction du nombre d'appels écoulés et de la capacité des centres de commutation).

Le système de décision adopté est un outil général de diagnostic avec rejet d'ambiguïté appelé *arbre de neurones*. Il s'agit d'un ensemble de neurones et d'un arbre de décision binaire (figure 2.6). Chaque neurone N_i comporte n entrées x_j du vecteur forme $\mathbf{x}$. Chaque entrée x_j est pondérée par un poids synaptique w_{ij}. La sortie $y_i(\mathbf{x})$ est calculée selon une fonction

d'activation f et un seuil w_{i0}. L'arbre de décision est un arbre binaire. Chaque nœud interne est associé de façon unique à un neurone. Les nœuds terminaux (feuilles) représentent les classes de pannes. Une branche de l'arbre est choisie en fonction de la sortie du neurone associé au nœud parent. Si la sortie $y_i(\mathbf{x})$ est telle que $y_i(\mathbf{x}) \in [S_i^1, S_i^2]$, il y a ambiguïté, les deux branches sont choisies. La figure 2.6 représente l'arbre de décision pour le centre de transit mixte de Nantes : il contient 8 classes de pannes.

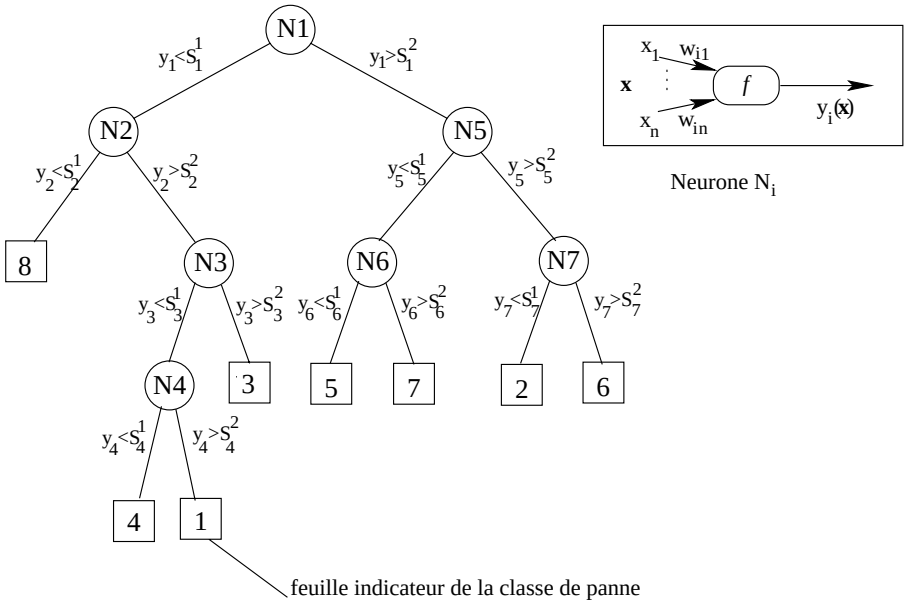

FIG. 2.6 – Arbre de décision pour le centre de transit de Nantes

La construction d'un arbre de neurones nécessite un ensemble d'apprentissage constitué de vecteurs formes étiquetés par la classe de pannes à laquelle ils appartiennent, autrement dit un ensemble de signature d'observations pour lesquelles on connaît le *type* de panne qui les a produites. Cette construction s'effectue en séparant les exemples de l'ensemble d'apprentissage en plusieurs sous-ensembles disjoints devant contenir chacun des vecteurs formes appartenant à la même classe de pannes. Cette disjonction s'effectue par la recherche de frontières linéaires entre ces sous-ensembles, les pondérations w_{ij} du neurone N_i étant déduites de l'équation linéaire représentant la frontière F_i. Les seuils S_i^1 et S_i^2 sont établis en fonction de la frontière F_i et déterminent la zone d'ambiguïté associée à F_i.

2.3.4 Reconnaissance de scénarios

Dans les approches précédentes, les systèmes de corrélation d'alarmes considèrent les observations comme un ensemble d'événements, sans relations temporelles entre ces observations. Autrement dit, pour ces systèmes de corrélation, le diagnostic de pannes proposé est identique que les observations soient reçues dans un ordre ou un autre avec des délais différents ou non. Dans certains systèmes, cette information temporelle est importante car elle peut per-

mettre de discriminer différentes explications. Le formalisme des *chroniques*, encore appelées *scénarios*, est adapté à la prise en compte de ces *contraintes temporelles*.

2.3.4.1 Modèle de chronique

Un *modèle de chronique* est constitué d'un ensemble d'observations et d'un ensemble de contraintes temporelles entre les instants d'occurrences de celles-ci. Cet ensemble de contraintes temporelles est représenté sous la forme d'un *graphe d'instants* (voir figure 2.7).

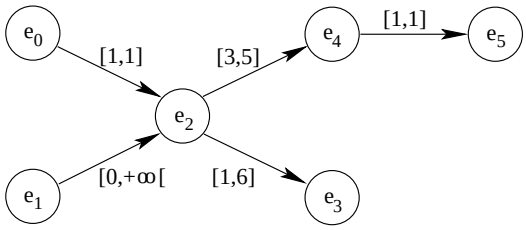

FIG. 2.7 – Graphe des instants d'un modèle de chronique

Dans l'exemple de la figure 2.7, chaque événement e_i est associé à une occurrence parmi les événements observés. La transition du graphe étiquetée $[t_{min}, t_{max}]$ entre un événement e_i et un événement e_j représente la contrainte temporelle « si e_i survient à la date t_i, alors e_j survient à la date t_j telle que $t_i + t_{min} \leq t_j \leq t_i + t_{max}$ » [1].

2.3.4.2 Reconnaissance de chroniques

Le diagnostic d'un système à l'aide de chroniques est fondé sur la reconnaissance *en ligne* de modèles de chroniques. Le principe de la reconnaissance d'une chronique est le suivant. À chaque observation, on maintient un ensemble de chroniques candidates, il s'agit d'un ensemble d'instances de modèle de chroniques pour lesquelles l'ensemble des observations reçues à cet instant est compatible avec le graphe des instants de chaque chronique. À la réception d'une nouvelle observation, on élimine de cet ensemble les chroniques qui ne sont pas compatibles (typiquement, la nouvelle observation est reçue trop tard ou trop tôt suivant ce scénario) et on ajoute les chroniques qui peuvent débuter avec l'arrivée de cette nouvelle observation. Une chronique est *reconnue* lorsque tous les événements représentés dans le graphe des instants ont eu lieu dans l'ordre et les délais notifiés dans le graphe. Une fois qu'une chronique est reconnue, l'information de diagnostic associée à cette chronique est notifiée.

Des outils de reconnaissance de chroniques mettent en œuvre ce principe (IxTeT [MA 90] [Dousson et al. 93], son successeur CRS (*Chronicle Recognition System*)). L'objectif principal est l'efficacité en ligne de la reconnaissance. CRS est un outil plus spécialisé pour la reconnaissance de chroniques dans les réseaux de télécommunications en proposant un enrichissement du formalisme des modèles de chroniques, notamment la possibilité de définir des scénarios génériques du type « occurrence de n événements du même type dans un intervalle donné ».

[1] Les intervalles peuvent ne pas contenir les bornes, dans ce cas évidemment, les relations temporelles sont strictes.

Cette extension permet en particulier de modéliser le phénomène de la redondance d'alarmes et donc servir de filtrage d'alarmes. La reconnaissance par chronique est une problématique qui est arrivée à maturité, certains industriels comme la société Ilog commencent à intégrer ce formalisme dans leurs produits [Berstel 02].

2.3.5 Inconvénients

Le principal inconvénient des approches à base de corrélations d'alarmes décrites ci-dessus est l'acquisition des règles de corrélations (séquences d'alarmes corrélées, étiquetage de vecteurs formes de l'ensemble d'apprentissage, chroniques...). Elles sont en effet fondées sur une connaissance de surface du système qui demande une expérience, une expertise. Afin de diminuer cette dépendance à l'expertise du système, des approches ont été développées afin d'acquérir automatiquement ces règles de corrélations. Concernant les chroniques par exemple, il existe deux techniques d'acquisition. Dans [Dousson et Du'o'ng 99] [Du'o'ng 01], les auteurs présentent un outil d'acquisition de chroniques FACE (*Frequency Analyser for Chronicle Extraction*) qui analyse les journaux d'alarmes. Les chroniques établies sont le résultat d'une recherche des *régularités* (scénarios fréquents) dans les journaux. Ces chroniques sont intéressantes car elles modélisent le comportement régulier du système. Il reste le problème de la sémantique associée à une chronique reconnue. La sémantique que l'on peut en effet établir n'est pas très explicative s'il n'y a pas un expert qui soit en mesure de dire que tel scénario régulier d'observations correspond à tel scénario de panne. La deuxième approche pour l'acquisition est une approche à base de modèle issue du projet Gaspar [Bibas et al. 96],[Rozé 97a],[Mayer 99],[Osmani 99]. L'idée est d'acquérir les chroniques par un apprentissage fondé sur la simulation d'un modèle de comportement du système à superviser (voir section 2.4.2.5).

2.3.6 Conclusion

Les systèmes de corrélation d'alarmes sont des outils puissants pour la reconnaissance *en ligne* de situations connues. L'acquisition des règles de corrélation demandent une expertise moindre que pour l'élaboration d'un système expert. Par contre, sans cette connaissance experte, il est très difficile d'attribuer une véritable explication (un diagnostic de pannes) à une situation observée et reconnue. Les systèmes d'acquisition automatique de règles de corrélations se basent plus sur les observations du système et moins sur la structure et le comportement du système, ce qui produit un système de règles de corrélations peu intuitives et peu explicatives (en particulier, dans les approches à base de reconnaissance de forme où il est difficile d'attribuer une explication à ce qu'un neurone a appris).

Les réseaux de télécommunications sont des systèmes complexes qui évoluent rapidement. Il devient donc nécessaire d'acquérir l'information utile au diagnostic de façon méthodique et automatique, en évitant autant que possible l'intervention d'un expert de la supervision de tel ou tel système. De plus, les opérateurs sont aussi bien intéressés par la panne primaire qui a causé telle séquence d'alarmes que par la propagation de cette panne à travers le réseau. Pour établir une véritable explication sur la propagation de pannes dans un tel système, on doit se baser sur des informations de plus bas niveau que les connaissances utilisées par les systèmes

de corrélation d'alarmes.

2.4 Diagnostic à base de modèles

La méthode, dite du *diagnostic à base de modèles* , ou encore du diagnostic à partir des principes premiers, a vu le jour aux États-Unis au milieu des années soixante-dix et a été formalisée au début des années quatre-vingts. Un nombre croissant de travaux ont été menés depuis et cette problématique est devenue un domaine de recherche à part entière de l'intelligence artificielle. Les articles les plus marquants sur ce domaine et publiés avant 1991 sont regroupés dans [Hamscher et al. 92]. On peut également retrouver la théorie logique du diagnostic à base de modèles dans le chapitre 1 de [sldd 01] rédigé par P. Dague.

2.4.1 Principes

Dans le domaine du diagnostic à base de modèles, le terme de *modèle* est employé par opposition à la connaissance associationniste de nature généralement empirique utilisée dans les méthodes traditionnelles de diagnostic en intelligence artificielle. Les connaissances incluses dans ces modèles décrivent la *structure* du système à diagnostiquer et son *comportement*. Le modèle structurel décrit généralement les liens, les connexions entre les différents composants du système à diagnostiquer. Quant au modèle comportemental, il est généralement le résultat de la *composition* des modèles de comportement de chaque composant du système. Le cadre théorique logique permettant de formuler rigoureusement le problème du diagnostic à base de modèles a été établi dans [Reiter 92] [dKleer et al. 92]. Voici en particulier les définitions associées à la notion de modèle.

Définition 2.1 (Modèle de système) *Un* modèle de système *est une paire* $(DS, COMPS)$ *où* DS*, la description d'un système, est un ensemble de formules de la logique des prédicats du premier ordre avec égalité et où* $COMPS$*, les composants de ce système, est un ensemble fini de constantes.* □

$COMPS$ décrit l'ensemble des composants du système à diagnostiquer et DS décrit le comportement des composants ainsi que la structure du système. Les *observations* sont définies comme suit :

Définition 2.2 (Ensemble d'observations) *Un* ensemble d'observations OBS *est un ensemble de formules du premier ordre avec égalité.* □

Ces deux définitions permettent de définir un *système observé* :

Définition 2.3 (Système observé) *Un modèle de* système observé *est un triplet* $(DS, COMPS, OBS)$ *où* $(DS, COMPS)$ *est un modèle de système et* OBS *un ensemble d'observations.* □

Seules les connaissances structurelles sont spécifiques aux systèmes en question ; les connaissances comportementales, ici plus ou moins directement liées à des lois de la physique, à des spécifications, sont généralement génériques et ne dépendent que du domaine choisi et non pas du système lui-même. Cette connaissance sur les comportements est donc réutilisable pour tout système du domaine en question et décomposable en bibliothèque de modèles comportementaux de composants génériques.

2.4.1.1 Diagnostic de cohérence

Une autre caractéristique du diagnostic à base de modèles est qu'il n'est nul besoin de savoir quoi que ce soit *a priori* sur les défauts ou dysfonctionnements pouvant affecter un système pour pouvoir le diagnostiquer : modéliser le comportement correct est suffisant. L'idée fondamentale est de comparer le comportement réel du système tel qu'il peut être observé par l'intermédiaire de capteurs et son comportement attendu tel qu'il peut être prédit grâce aux modèles de bon comportement. Le résultat de cette comparaison permet d'établir un *diagnostic de cohérence*. Si ces modèles sont corrects, en ce sens qu'ils sont effectivement vérifiés par un système en bon fonctionnement, toute contradiction entre les observations et les prédictions déduites des modèles est nécessairement la manifestation d'un dysfonctionnement, c'est-à-dire de la présence d'un ou plusieurs défauts. Dans le cadre de la théorie logique, DS mentionne un prédicat unaire $AN(x)$ où $x \in COMPS$ et qui est interprété comme signifiant anormal. Si, pour $\Delta \subseteq COMPS$, on note $D(\Delta) = (\wedge AN(c) | c \in \Delta) \wedge (\wedge \neg AN(c) | c \in COMPS - \Delta)$, le diagnostic est défini par :

Définition 2.4 (Diagnostic de cohérence) *Soit $(DS, COMPS, OBS)$ un système observé, son* diagnostic *est un $D(\Delta)$ avec un $\Delta \subseteq COMPS$ tel que :*

$$DS \cup OBS \cup \{D(\Delta)\} \text{ est satisfiable.}$$

□

Ce type de raisonnement par l'absurde fait qu'un défaut est par définition n'importe quoi d'autre que le comportement attendu et il n'est pas recensé parmi une liste finie prédéterminée. Cette approche est une méthode de raisonnement très puissante, qui pallie la plupart des limites des approches traditionnelles et qui peut être logiquement fondée : la détection de dysfonctionnements par réfutation du bon comportement prédit est un raisonnement logiquement correct, ce que n'est pas le cas de la détection de dysfonctionnements par corroboration avec un mauvais comportement prédit.

Le diagnostic à base de modèles, dans ses principes de base, traite principalement de la tâche de localisation des défauts et également, après extension, de celle d'identification de ces défauts.

2.4.1.2 Utilisation d'un modèle de dysfonctionnement

L'idée première du diagnostic à base de modèles est de se passer des connaissances sur les défauts ou les dysfonctionnements. Mais certaines connaissances de ce type, si elles sont disponibles, peuvent aider à la localisation des défauts et il serait dommage de s'en passer.

De plus, il est généralement indispensable d'avoir de telles connaissances si l'on souhaite identifier les défauts après les avoir localisés. C'est pourquoi des extensions du formalisme ont été proposées, qui permettent d'exprimer des modèles de dysfonctionnement : GDE+ [Struss et Dressler 92] et Sherlock [dK et Williams 92]. Il est important de remarquer que, contrairement aux approches traditionnelles, où les relations entre défauts et symptômes sont empiriques, spécifiques au système à diagnostiquer et ne peuvent en aucun cas garantir la validité logique du diagnostic établi, cette validité demeure garantie dans les extensions en question. Remarquons que l'on peut toujours introduire en plus, si besoin est, une telle connaissance empirique, mais uniquement cette fois au titre d'heuristiques aidant à parcourir éventuellement plus vite l'espace de recherche conduisant aux solutions mais sans influence sur celles-ci, qui reposent ici sur des principes logiques rigoureux. Au lieu de n'avoir comme précédemment que deux modes de comportement par composant, correct et incorrect, dont seul le premier est modélisé, s'ajouteront cette fois aux modes corrects plusieurs modes de dysfonctionnements (en général deux à deux exclusifs) modélisés. Mais pour tenir compte de l'impossibilité d'une énumération exhaustive de tous les défauts possibles, il sera toujours ajouté un mode inconnu dépourvu de tout modèle qui est censé regrouper tous les comportements défectueux non répertoriés.

2.4.1.3 Diagnostic abductif

Le premier objectif du diagnostic est de détecter/localiser voire identifier un dysfonctionnement à partir des observations du système. Mais parfois, il peut être intéressant que le diagnostic *explique* les observations. Dans ce cas, il faut passer à un raisonnement de type *abductif* où l'on cherche les causes qui expliquent les symptômes. En l'absence de modèle de comportements défectueux, le pouvoir explicatif est nul. Le diagnostic est uniquement fondé sur la restauration de la cohérence avec les observations. Pour établir un diagnostic abductif, il faut pouvoir disposer d'un modèle de dysfonctionnement du système. Formellement,

Définition 2.5 (Diagnostic abductif) *Soit* $(DS, COMPS, OBS)$ *un système observé et* $OBS = E \cup S$ *une partition de* OBS*,* S *correspondant aux observations que l'on veut expliquer. Un* diagnostic abductif *pour* $(DS, COMPS, E \cup S)$ *est un* $D(\Delta)$ *avec* $\Delta \subseteq COMPS$ *tel que :*

$$DS \cup E \cup \{D(\Delta)\} \text{ est satisfiable et } DS \cup E \cup \{D(\Delta)\} \models S.$$

□

Ici, on partage les observations OBS en deux sous-ensembles distincts E et S, et l'on cherche dans ce cadre les modes comportementaux qui, d'une part sont cohérents avec les observations E, et d'autre part *impliquent* S conjointement avec E. En faisant varier à volonté E et S, on a ainsi tout un spectre qui s'étend du diagnostic purement fondé sur la cohérence (cas où $S = \emptyset$) au cas du diagnostic purement abductif (cas où $E = \emptyset$) [Console et Torasso 92].

2.4.1.4 Diagnostic à base de modèles et supervision

La théorie du diagnostic à base de modèles est à caractère atemporel. Plus précisément, la tâche du diagnostic consiste, à partir d'un système, éventuellement dynamique, en dysfonc-

tionnement, à localiser et si possible identifier les composant défectueux responsables de ce dysfonctionnement. Dans ce cadre théorique, les défauts sont supposés présents au début du processus de diagnostic et permanents durant tout ce processus. Ces hypothèses ne sont bien sûr plus satisfaites dans le cas du diagnostic en ligne, les paramètres même du système peuvent varier au cours du temps, traduisant en particulier l'apparition et l'évolution d'un défaut. Or le diagnostic en ligne, et plus généralement l'activité de supervision dans laquelle il s'insère, est du point de vue de la sûreté et sur le plan économique, d'une importance considérable.

L'idée de base des travaux sur le diagnostic en ligne à base de modèle reste la même que pour le diagnostic hors ligne : comparer le comportement prédit à base de la modélisation et celui réellement observé. Toute discordance entre les deux indiquera la présence d'au moins un défaut, que l'on cherchera à localiser et identifier par l'examen du chemin déductif qui a conduit à la contradiction. Outre les problèmes de temps réel qui peuvent être cruciaux pour des processus à dynamique rapide lors de défaillance brusque et la nécessité des techniques de compilation hors ligne, la problématique du diagnostic en ligne introduit une nouvelle composante temporelle : il faut à présent synchroniser les temps de prédiction et d'observations. En effet, dès que le temps de réponse de la dynamique propre du système ne peut être considéré comme négligeable, on ne peut plus effectuer le diagnostic sur des photographies instantanées du système, comme une simple succession de diagnostics statiques. Le logiciel de diagnostic doit intégrer la dynamique du système ainsi que la dynamique des défauts. Ceci soulève des problèmes beaucoup plus difficiles que le diagnostic hors ligne : suivi temporel du processus évolutif, apparition de nouveau dysfonctionnement (et disparition). À l'évolution naturelle du système, de sa dynamique propre se superpose sa modification due à l'apparition et à l'évolution des pannes.

Dans le cadre d'un système muni d'un système de supervision au sein même de l'installation, il faut alors prendre en compte des problèmes d'incohérence entre capteurs, de chemin et de délai de rapatriement des messages, d'ordre chronologique d'arrivées des messages qui peut être différent de celui de leurs émissions, de masquage et de perte de messages, etc (cf section 1.4.5.2). C'est tout le problème de la gestion d'alarmes au niveau du superviseur qui nécessite une modélisation à la fois de l'installation supervisée et du système de supervision pour la génération d'explications et de critères décisionnels de haut niveau sémantique et de crédibilité suffisante à l'attention des opérateurs de supervision.

2.4.2 Travaux sur les réseaux

De nombreux travaux ont déjà été développés dans le domaine du diagnostic de pannes dans les réseaux de télécommunications par l'utilisation des principes énoncés ci-dessus. La différence notable entre ces différentes approches est la granularité du modèle qui se répercute sur l'information donnée en résultat du système de diagnostic.

2.4.2.1 Utilisation d'un graphe de dépendances

Le modèle utilisé dans [Bouloutas et al. 92], [Gruschke 98b], [Gruschke 98a] est un *graphe de dépendances*. Chaque nœud de ce graphe correspond à un objet qui peut potentiellement subir des pannes. Typiquement dans le cadre des réseaux de télécommunications, il peut s'agir

des objets de gestion associés aux éléments du réseau (voir section 1.3.2). Néanmoins, il peut s'agir aussi d'abstractions de ces objets (groupe d'objets) ou encore des objets tels que des connexions (virtuelles ou non). Chaque arc du graphe représente une dépendance entre deux objets. Autrement dit, si une panne survient sur l'objet cible d'un arc, cela peut éventuellement influer sur le comportement de l'objet source (et donc l'objet source dépend de l'objet cible). Ce modèle permet ainsi d'exprimer le phénomène de propagation de pannes et de la cascade d'alarmes qui peut en découler.

Ce modèle est utilisé pour construire un système de corrélation (voir section 2.3) [Gruschke 98b]. L'algorithme de corrélation procède comme suit. Pour chaque alarme, il déduit l'objet qui est l'émetteur et donc celui qui est la cause de l'alarme. Ensuite, l'algorithme recherche dans le graphe de dépendances les objets qui peuvent influer sur le comportement de l'objet émetteur de l'observation. Cette recherche étant effectuée pour chaque objet émetteur d'un événement observé, l'algorithme retourne un ensemble d'objets qui, par influence, expliquent l'ensemble des observations. Cet ensemble d'objets est l'information corrélée. Cet ensemble d'objets est aussi appelé *domaine d'alarmes* [Bouloutas et al. 92]. Selon ce modèle, les alarmes corrélées sont les alarmes dont l'intersection des domaines est non-vide : un tel ensemble d'alarmes est aussi appelé *grappe d'alarmes* (traduction de *alarm cluster*) .

L'avantage majeur de ce modèle est son acquisition. En effet, l'information de dépendance peut s'extraire du modèle de gestion. En particulier, les dépendances de comportements sont liées à la topologie des objets modélisés. De plus, si le système évolue, le modèle est facilement adaptable afin que le système de corrélation soit toujours opérationnel.

Ce système de corrélation a pour objectif de répondre à cette question : étant donné cet ensemble d'observations, quels sont les facteurs communs qui en sont la cause ? Autrement dit, ce système ne gère pas les pannes multiples et indépendantes pouvant avoir lieu en même temps. De plus, ce système localise la source commune des problèmes mais n'identifie pas le problème dans un cadre général (les auteurs considèrent qu'une observation émise par un objet est assez pertinente pour l'associer à un problème unique sur cet objet). Étant donnée la nature du système diagnostiqué, tout objet peut influer sur tout autre (graphe de dépendances connexe) si bien que le résultat de la corrélation peut être l'ensemble des objets, ce qui est peu pertinent en général.

Pour résoudre ce dernier problème, [Katzela et al. 95] ajoute des pondérations sur les nœuds et sur les transitions du graphe de dépendances. Un poids p_i sur un nœud n_i correspond à la probabilité que l'objet représenté par n_i puisse tomber en panne indépendament des autres (autrement dit, la probabilité de l'occurrence d'une panne primaire sur cet objet). De même, le poids p_{ij} sur la transition entre un nœud n_i et un nœud n_j correspond à la probabilité que n_i puisse tomber en panne sachant que n_j est en panne (probabilité de panne secondaire). L'objectif de ces pondérations est double. Premièrement, la définition du domaine d'une alarme peut être restreinte en ne considérant que l'ensemble des objets pouvant influencer avec au moins une certaine probabilité. Deuxièmement, ceci permet de définir un « meilleur » domaine pour une grappe d'alarmes : il s'agit du sous-ensemble d'objets (nœuds) avec les propriétés suivantes :

– chaque objet a une influence sur toutes les alarmes de la grappe ;

– la probabilité qu'au moins un des objets soit atteint par une panne primaire est maximale.

Le système de diagnostic est donc un algorithme de recherche de cet ensemble minimal.

[Katzela et al. 95] a prouvé néanmoins que cette recherche était un problème NP-complet, et donc l'usage d'heuristiques pour trouver une solution approchante est nécessaire.

2.4.2.2 Diagnostic de cohérence vu comme un problème de satisfaction de contraintes

[Riese 93b] présente un algorithme pour le diagnostic de protocoles de réseaux (HMDP : *Heuristic Model-based Diagnosis of communication Protocols*). Le modèle de diagnostic est décrit à l'aide de transducteurs étendus par des conditions de gardes et des conditions temporelles. Ces modèles décrivent le comportement nominal des protocoles et l'objectif du diagnostic est de détecter les composants défectueux (diagnostic de cohérence). La simulation de ce modèle afin d'en mesurer les écarts avec les observations est définie comme un problème de satisfaction de contraintes. Une variable V_i est associée à chaque observation O_i (une observation est un événement daté) et le domaine de ces variables est l'ensemble des transitions du transducteur modélisant le protocole. Les contraintes entre les variables sont établies à partir du modèle. Par exemple, une transition affectée à V_i a un label contenant forcément un message du même type que celui de O_i (contrainte unaire). Autre exemple, des contraintes entre deux variables V_i et V_j sont définies s'il existe un chemin de transitions entre V_i et V_j (ce sont des contraintes de prédictions, V_i « prédit » V_j). L'ensemble de ces contraintes permet d'établir un réseau de contraintes entre les variables. Le problème de diagnostic devient alors un problème de satisfaction de contraintes : trouver des valeurs pour les variables V_i telles que les contraintes soient satisfaites. Si une telle solution existe, cela signifie qu'il existe au moins un chemin de transitions dans le modèle qui explique le comportement observé : le comportement observé est compatible avec la prédiction du modèle. Dans le cas contraire, il y a un écart entre les comportements prédit et observé, ce qui traduit la détection d'une panne. L'algorithme HMDP va alors tenter via des heuristiques de modifier le modèle initial afin que les contraintes soient effectivement vérifiées (ajout de transitions, modification de conditions de garde ou d'intervalles temporels). Ces modifications, réduisant l'écart entre le comportement prédit et le comportement observé, constituent une nouvelle information de diagnostic servant à mieux identifier l'erreur qui s'est produite.

Les avantages de cet algorithme sont multiples. Puisqu'il s'agit de protocoles, il existe de nombreux formalismes dans lesquels ils sont spécifiés, ils sont en particulier spécifiés à l'aide de systèmes de transitions tels que les transducteurs. L'algorithme HMDP opère sur le réseau de contraintes extrait du modèle étudié et donc il en est assez indépendant. Étant donné que la recherche d'une solution consiste à vérifier qu'il existe des chemins de transitions compatibles avec le comportement observé, l'algorithme est en mesure de générer des hypothèses de pannes multiples, de plus cet algorithme gère l'incomplétude ou l'incertitude liées aux observations [Riese 93a]. Cet algorithme est donc bien adapté pour le diagnostic de protocoles réseaux.

Deux problèmes se posent néanmoins. Le premier est le temps de calcul de la réduction des écarts qui ne peut pas se concevoir de manière en ligne. Deuxièmement, dès que l'on considère des systèmes dynamiques plus importants, l'algorithme est confronté au problème de la taille du modèle. En effet, HDMP nécessite d'extraire un réseau de contraintes à partir d'un *modèle global* du système, ce qui est une limitation si ce modèle n'est pas implantable (trop de transitions possibles) : [Riese 93c] montre par exemple que l'algorithme HDMP est efficace si l'on met en place une structure de données dont la taille est proportionnelle au

nombre de transitions du modèle ce qui n'est pas possible dans le cas qui nous intéresse.

2.4.2.3 Utilisation d'un modèle causal

[Kehl et al. 93] [Bigham et al. 92] [Azarmi et al. 93] présentent un outil de maintenance générique pour les réseaux de télécommunications : GMS (*Generic Maintenance System*). Cet outil a pour objectif de corréler les alarmes, d'effectuer un diagnostic et d'estimer une procédure de réparation. GMS est fondé sur un modèle causal qui a la particularité d'être modulaire. Ce modèle est en effet constitué d'entités fonctionnelles (modèle comportemental) qui sont connectées les unes avec les autres (modèle structurel). Chaque entité fonctionnelle est constituée d'un ensemble de règles de cause à effets : les causes peuvent provenir d'une autre entité fonctionnelle (dans ce cas, il y a donc une connexion entre ces deux entités) et impliquent des effets sur l'*état de service* de l'entité fonctionnelle courante.

Le calcul du diagnostic est établi en plusieurs phases. La première est un raisonnement abductif : étant donné un ensemble d'observations (des symptômes considérés comme des effets), cette première phase consiste à établir les causes pouvant expliquer ces effets en « remontant les règles » décrites dans le modèle. Ayant trouvé un ensemble de causes candidates, la deuxième phase consiste à détecter si toutes les conséquences de ces causes sont cohérentes avec les observations : cette phase de déduction revient à *simuler le modèle* à partir des causes candidates. La troisième phase est celle de tests (*polling*) afin de vérifier les explications déduites par la deuxième phase. Les résultats de ces tests constituent de nouveaux symptômes qui peuvent éventuellement servir à affiner le diagnostic en réitérant les différentes phases précédemment décrites.

Cette approche est intéressante du point de vue de la modélisation. Le réseau est représenté par deux types de modèles :

- un *modèle structurel* : il décrit les *interactions* entre les différentes entités, dans ce système ces interactions sont de type causal ;
- un *modèle comportemental* : il décrit le comportement de chaque entité, ici ce comportement est un ensemble de causes conduisant à des effets.

L'intérêt d'une telle modélisation est la modularité. Si le système évolue (des composants sont modifiés), il est aisé de modifier les entités fonctionnelles correspondantes ainsi que les interactions avec le voisinage. L'information causale est très riche si bien que le calcul de l'identification d'une panne est plus efficace que celui détaillé dans la section 2.4.2.2. Par contre, l'information causale est plus difficile à acquérir car elle n'est pas ou peu décrite à travers les normes de protocoles et peut demander une expertise (voir section 2.2.3). De plus, les réseaux de télécommunications étant constitués de composants très interactifs, le modèle causal est donc important : les phases d'abduction et de déduction nécessaires aux calculs des explications ne sont pas efficaces dans le sens où elles ne peuvent être utilisées en ligne.

Des travaux plus récents utilisent un modèle un peu similaire [Chirashnya et al. 01b] [Chirashnya et al. 01a]. L'application privilégiée dans ce cas est un réseau commuté (SAN : *Switch Area Network*). L'objectif dans ce cas est d'établir des recommandations pour le remplacement d'un composant en fonction des mauvais fonctionnements diagnostiqués. Ce diagnostic est établi au niveau de la couche transport du réseau et donc des paquets d'informations transmis aux différents composants. Chaque composant du réseau est représenté par une en-

tité pouvant être sujette à des mauvais fonctionnements dont la conséquence est l'émission d'alarmes et l'envoi de paquets d'information au voisinage. L'information utilisée est moins abstraite que dans le cas de GMS et n'est établie que par rapport à une seule couche réseau (la couche transport). Ceci facilite donc l'acquisition de l'information (venant des normes ou des caractéristiques données par le constructeur de l'équipement) de plus, la modularité permet de prendre en compte les évolutions du système. Dans cette application, on considère aussi que les mauvais fonctionnements sont munis d'une probabilité d'occurrences et qu'il y a une probabilité de pertes d'alarmes. Lors du traitement d'une alarme, la phase d'abduction construit donc un réseau Bayésien. Contrairement à GMS, la phase de déduction ne consiste pas à éliminer les inconsistances liées aux observations mais consiste à établir quelles sont les alarmes attendues, ces occurrences étant munies d'une probabilité d'avoir été perdues.

2.4.2.4 Utilisation d'un modèle de dysfonctionnement

L'une des particularités des applications telles que les réseaux de télécommunications est qu'il existe des spécifications de fonctionnements d'équipements dans le cas normal de fonctionnement mais aussi dans des modes dégradés. Il est plus aisé d'établir des modèles dans lesquels non seulement on y décrit le comportement nominal mais aussi le fonctionnement en cas de pannes, en cas de dysfonctionnements du réseau. C'est partant de ce constat que des travaux ont été effectués en vue d'établir des algorithmes fondés sur ce type de modèle.

[Sampath et al. 95], [Rozé et Cordier 98], [Aghasaryan et al. 98], [Baroni et al. 99] ont mis en place de tels modèles pour le diagnostic sur des systèmes à événements discrets. Ces modèles décrivent le comportement en cas de panne du système, si bien que le diagnostic est établi en retrouvant dans le modèle l'ensemble des événements pouvant avoir eu lieu et qui expliquent les observations. Ces ensemble d'événements (comportements) ainsi obtenus peuvent contenir des pannes ou non et permettent ainsi d'en déduire des hypothèses de diagnostic. Suivant les auteurs et les applications étudiées, les formalismes varient mais l'objectif final est la description des comportements par un système de transitions. Néanmoins dans ces modèles, on retrouve certaines constantes :

- *réactivité :* les systèmes sont modélisés comme des systèmes réagissant à des pannes. Ces pannes peuvent être spontanées (*primaires*) ou le résultat d'une propagation de pannes (par échange de messages par exemple) (*pannes secondaires*). La deuxième caractéristique est que l'on modélise les comportements liés à la propagation des pannes (les échanges de messages liés à des protocoles...).
- *modularité :* les systèmes sont modélisés de façon modulaire, à l'aide d'un modèle comportemental et d'un modèle structurel. Le *modèle structurel* décrit la manière dont les différents composants du système communiquent entre eux (échanges de messages, partage de ressources...). Le modèle comportemental, quant à lui, décrit le *comportement local* de chaque sous-système (réaction à des pannes, à des réceptions de messages, production d'observations...). Le *comportement global* du système est alors implicitement représenté par ces deux types de modèles. Puisque ces modèles sont des systèmes de transitions, il est toujours possible de définir une *loi de composition* permettant d'établir explicitement ce comportement global [Arnold 92].

À partir d'une telle modélisation, on peut distinguer deux types d'approches pour le diag-

nostic de systèmes :

1. *les approches hors ligne :* ce sont des approches dont l'objectif est de répondre à un problème de diagnostic étant donné un ensemble d'observations connu *a priori*. Dans une telle approche, on considère que le temps de réponse à un problème n'est pas limité.

2. *les approches en ligne :* ce sont des approches ou l'on ne connaît pas *a priori* l'ensemble complet des observations. L'objectif ici est de donner un diagnostic en fonction des observations reçues et de l'adapter en fonction d'éventuelles nouvelles observations. Cette adaptation doit être efficace car il faut être en mesure de suivre la réception des observations.

2.4.2.5 Approches hors ligne

Dans les travaux de [Baroni et al. 98][Baroni et al. 99], les auteurs étudient le problème du diagnostic sur une classe de systèmes à événements discrets appelés *systèmes actifs*. Ces systèmes sont représentés à l'aide d'un ensemble d'automates communicants. Les communications sont établies à l'aide d'échanges de messages sur des canaux de communications. Un canal est représenté par une file de messages bornée pouvant avoir différentes politiques dans le cas où la file est saturée (perte du dernier message, écrasement du premier, ...). Étant donnée une séquence d'observations, l'objectif est d'établir à partir des automates communicants et des files de messages, l'ensemble des comportements du système pouvant expliquer cette séquence d'observations. Cette construction de comportements est établie de façon modulaire, elle dépend des opérations décrites par les transitions des automates et des messages contenus dans les canaux de communications. Dans [Baroni et al. 98], cette construction est établie en suivant un *plan de reconstruction*. Ce plan de reconstruction décrit les étapes de calcul du comportement en privilégiant le traitement commun de certains sous-ensembles de comportements locaux associés à des grappes de composants (*clusters*), avant la construction complète. Ce plan de reconstruction est établi en fonction des canaux de communications : il faut traiter en priorité les comportements locaux associés aux composants qui sont connectés à l'aide de tels canaux. Cette préférence vient du fait qu'elle permet d'éliminer plus rapidement des comportements locaux incompatibles avec les observations (la réception d'un message sur un composant impose qu'il ait été émis par un autre).

Dans le cadre du projet GASPAR (Gestion d'Alarmes par Simulation de PAnnes sur le Réseau), [Bibas et al. 96],[Rozé 97a],[Mayer 99],[Osmani 99] proposent une architecture pour le diagnostic où un tel modèle est aussi utilisé hors-ligne (voir figure 2.8).

Cette architecture est composée de quatre modules. Le module de modélisation sert à construire le modèle du réseau (utilisation d'automates communicants temporels [Rozé 97a]). Le module de simulation utilise ce modèle hors-ligne. L'objectif de ce module est de simuler des pannes afin d'établir les séquences d'alarmes produites si ces pannes se produisent sur le système. Le module de discrimination a l'objectif de produire à partir des résultats de la simulation des scénarios caractéristiques des différentes pannes pouvant se produire (sous forme de chroniques (voir section 2.3.4)). Le module de reconnaissance, quant à lui, est utilisé en ligne afin de récupérer les observations pour reconnaître les scénarios établis hors-ligne.

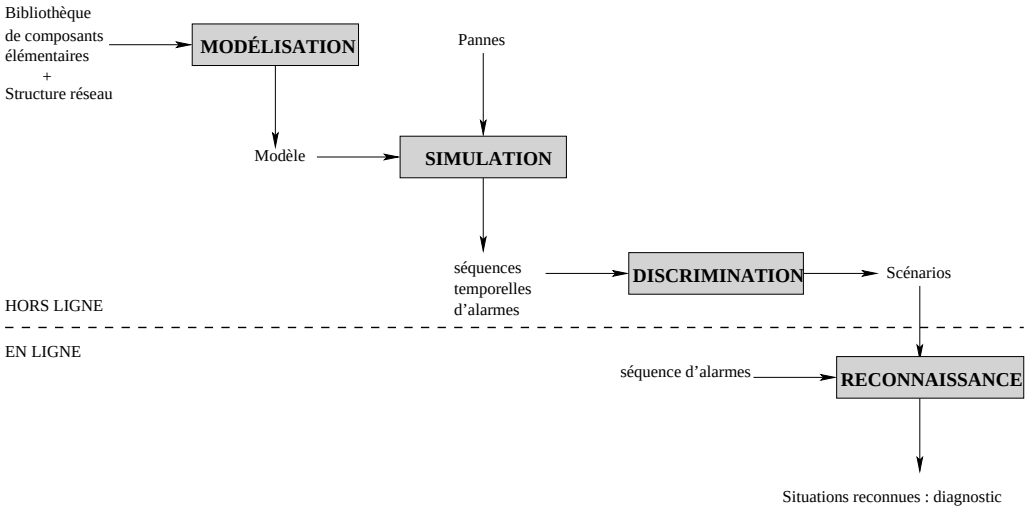

FIG. 2.8 – Architecture du projet Gaspar

2.4.2.6 Approches en ligne

Dans les approches en ligne, l'efficacité du calcul du diagnostic est primordiale. Afin de résoudre ce problème, deux voies sont possibles :

1. on compile le plus possible les informations de diagnostic hors ligne, afin de minimiser la tâche du processus en ligne de diagnostic ;
2. on fait une approximation du diagnostic en ne recensant que les diagnostics suivant des critères de vraisemblance, d'invraisemblance...

Compilation des informations de diagnostic : approche diagnostiqueur
[Sampath et al. 95][Sampath et al. 98] proposent une structure de données appelée *diagnostiqueur* (traduction de *diagnoser*). Le diagnostiqueur est un automate à nombre fini d'états, dont les événements déclenchant les transitions sont les événements observables du système et dont les états fournissent des informations sur les pannes ayant obligatoirement eu lieu.

Sur la figure 2.9, un exemple de diagnostiqueur est présenté (automate de droite). Ce diagnostiqueur est établi à partir du modèle global du système (automate de gauche, état initial *e1*). Les événements observables sont *o1* et *o2*. Le diagnostiqueur permet de suivre le comportement observable du système. À chaque état, on dispose d'une information de pannes. Dans l'état initial du diagnostiqueur, on apprend que le système est dans l'état *e1* et qu'aucune panne ne s'est produite. Si l'on observe *o1* alors le diagnostiqueur nous informe que le système est soit dans l'état *e4* et tout est normal, soit dans l'état *e3* en ayant subi la panne *p1*. Si on observe par la suite *o2*, le système est alors dans l'état *e3* : dans cet état on ne peut pas savoir si c'est *p1* ou bien *p2* qui a eu lieu, il y a ambiguïté.

Le calcul du diagnostiqueur est hors ligne et nécessite la construction d'un modèle global. Pour les systèmes représentés de façon modulaire (un ensemble de modèles comportementaux et un modèle structurel), le modèle global du système est établi en appliquant une opération

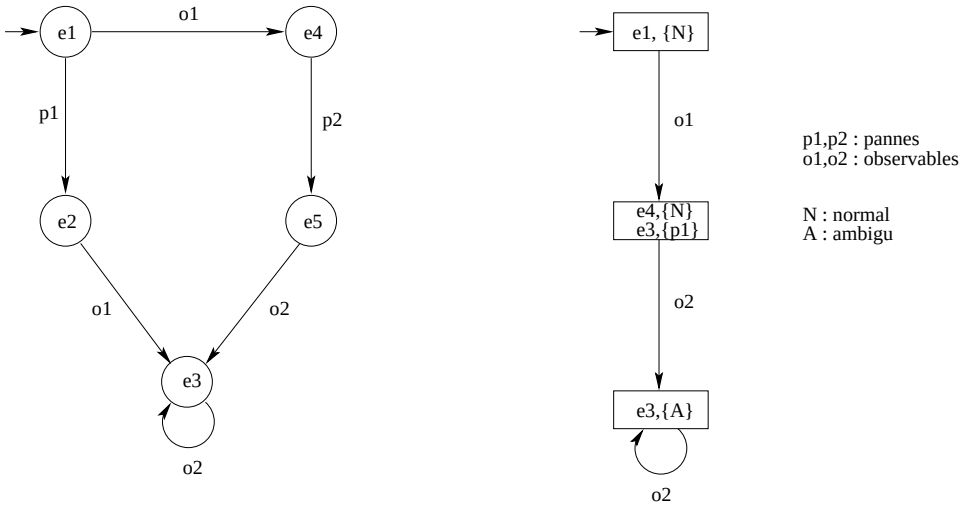

FIG. 2.9 – à gauche : modèle du système, à droite son diagnostiqueur

de *composition* sur l'ensemble des automates. Le processus de diagnostic, quant à lui, est en ligne et consiste à parcourir l'automate diagnostiqueur en fonction des observations reçues et de donner l'information résumée dans l'état courant du diagnostiqueur.

Des travaux concernant l'utilisation d'un diagnostiqueur dans le cadre des réseaux de télécommunications ont été développés dans [Rozé 97a]. On y propose en particulier une extension du diagnostiqueur pour prendre en compte une information temporelle sur la réception d'observations (délais entre deux observations). Néanmoins, un tel diagnostiqueur n'est pas exploitable dans le cadre des gros systèmes, en effet le modèle global du système n'étant pas implantable, on ne peut établir un tel diagnostiqueur. [Rozé et Laborie 98] et [Rozé et Cordier 02] proposent de régler ce problème en introduisant la notion de *diagnostiqueur générique*. Au lieu de construire le diagnostiqueur sur le modèle global, on le construit à partir d'un *modèle générique* : il s'agit d'un modèle factorisé du système, le système peut en effet être représenté par un ensemble d'instances de ce modèle générique. Le processus de diagnostic consiste dans ce cas à maintenir un ensemble d'hypothèses de diagnostic pour chaque instance du modèle en consultant le diagnostiqueur générique en fonction des observations reçues de chaque instance. Le problème de la généricité est qu'il n'est pas toujours possible de déterminer un modèle facteur du modèle complet du système.

Probabilisation du problème de diagnostic : algorithme de Viterbi

Une autre façon de rendre plus efficace le processus de diagnostic en ligne est d'en donner une approximation. [Aghasaryan 98] [Aghasaryan et al. 98] proposent l'utilisation d'un modèle de dysfonctionnement sur lequel est apposée une information sur la vraisemblance de tel ou tel comportement (réseau de Petri partiellement stochastique). L'objectif de diagnostic se résume alors à la recherche dans ce système de transitions des comportements les plus vraisemblables qui expliquent les observations reçues. Cette recherche revient donc à établir des chemins de transitions qui maximisent le critère de vraisemblance : cette recherche est effectuée à l'aide

d'un *algorithme de Viterbi* [Forney 73]. Le grand avantage de cette approche est que l'on peut rendre un résultat de diagnostic de façon très efficace. De plus, l'introduction d'un critère de vraisemblance permet de mieux gérer les problèmes liés à l'incertitude des informations disponibles (perte d'alarmes, incomplétude du modèle). Néanmoins cette approche a aussi certains inconvénients.

- Il faut établir un critère pertinent pour l'Opérateur. C'est un problème difficile. En effet, intuitivement, on pourrait penser que de donner le comportement le plus probable à l'Opérateur est satisfaisant. Néanmoins, un bon système de diagnostic se doit d'identifier des problèmes graves (nécessitant des interventions humaines par exemple) : ce genre d'hypothèses est généralement moins probable que d'autres moins graves et peut être « oublié » par le système de diagnostic.
- L'acquisition des informations stochastiques est difficile. Il n'existe pas d'information sur la vraisemblance d'un comportement dans les normes. Cette information de vraisemblance ne peut être établie que grâce à des experts du système étudié ou à des techniques d'entraînements de modèles nécessitant des journaux d'alarmes étiquetés (alarmes étiquetées avec l'explication de leur occurrence).

2.4.2.7 Les approches décentralisées

Les méthodes évoquées précédemment font l'hypothèse que l'information de diagnostic contenue dans le modèle est centralisée. Les systèmes tels que les réseaux de télécommunications sont des systèmes de grandes tailles, ainsi l'utilisation d'une information centralisée pour l'établissement d'un diagnostic en ligne peut s'avérer difficile voire impossible si le système est trop grand. Étant donnée la nature distribuée des systèmes tels que les réseaux de télécommunications, il est intéressant d'étudier des systèmes de diagnostic reposant sur cette architecture : ces systèmes sont des systèmes de diagnostics décentralisés. Dans cette approche, le système de diagnostic est constitué de plusieurs modules de diagnostic. Chaque module est responsable du diagnostic d'un sous-ensemble d'éléments du système et ne dispose que des informations liées à ce sous-ensemble. Les diagnostics produits par ces différents modules (des *diagnostics locaux*) doivent être ensuite combinés afin d'obtenir le diagnostic du système complet (le *diagnostic global*).

[Debouk et al. 98] [Sengupta 98] [Debouk et al. 00a] proposent une telle architecture. Le système supervisé est constitué de plusieurs sites. À chaque site est associé un *diagnostiqueur local* qui constitue l'information de diagnostic associée au site (fondée sur les observations reçues de ce site). En ligne, lorsqu'une observation est reçue, le système de diagnostic active le diagnostiqueur du site afin d'adapter le diagnostic local. Une fois l'adaptation effectuée, le système active un *protocole* entre les diagnostiqueurs afin qu'ils se mettent d'accord sur le diagnostic du système à proposer. L'inconvénient majeur de cette approche réside dans la construction des diagnostiqueurs locaux. Un diagnostiqueur selon [Debouk et al. 98] [Sengupta 98] [Debouk et al. 00a] est une adaptation du diagnostiqueur proposée par [Sampath et al. 95][Sampath et al. 98] qui nécessite la construction du modèle global du système : cette construction n'est pas possible pour les systèmes de grande taille.

[Aghasaryan et al. 98] [Fabre et al. 00] proposent également une telle architecture distribuée. Dans ce cas, la construction du diagnostic local ne nécessite pas la connaissance du

modèle global. La constitution du diagnostic du système s'effectue par un échange d'informations entre les différents sites de calcul du diagnostic (appelés des *joueurs* ou encore des *agents*). Ici, le *protocole* d'échange repose sur le critère de vraisemblance associé aux hypothèses locales de diagnostic afin d'établir les diagnostics du système les plus vraisemblables (voir section 2.4.2.6).

2.5 Synthèse, difficultés et besoins

De nombreux travaux de recherche en diagnostic de systèmes dynamiques ont déjà été effectués depuis de nombreuses années. Chaque technique relève d'une époque, des technologies utilisables et des ressources informatiques disponibles. Le diagnostic a tout d'abord été vu comme un ensemble de règles gérées par un système expert. Ces systèmes à connaissance de surface sont efficaces mais ils sont inadaptés dès lors que les systèmes supervisés évoluent (voir section 2.2). Des techniques moins fondées sur l'expertise sont alors apparues avec les systèmes de corrélations. L'information nécessaire, bien que restant liée à l'expertise, est plus faible et donc les systèmes de diagnostic sont plus faciles à maintenir. Le résultat de tels systèmes peut varier de la corrélation simple d'alarmes, à l'identification de la panne étant la cause d'un ensemble d'alarmes corrélées. Avec le diagnostic à base de modèles, un autre type de connaissance pour le diagnostic de système est apparu : les connaissances profondes. Les besoins se font sentir lorsque les systèmes supervisés sont plus complexes et qu'une explication plus profonde des observations est nécessaire.

Dans le cadre des réseaux de télécommunications, l'aide à l'interprétation des alarmes pour l'Opérateur de supervision consiste à *suivre le comportement observé* du système et à *établir un diagnostic* du système basé sur ces observations. Ce diagnostic consiste à mettre en évidence la présence d'une ou de plusieurs pannes dans le système à un instant donné, ces pannes pouvant disparaître. La présence d'une panne peut ne pas être détectable directement mais grâce à la détection de ses conséquences sur d'autres composants. Ce problème nécessite d'établir un modèle riche en informations permettant d'établir en ligne des hypothèses de propagation de pannes expliquant les observations. De plus, on doit prendre en compte le fait qu'un réseau de télécommunications évolue au cours du temps. Cela implique que le système de diagnostic doit être adaptable facilement en fonction de ces évolutions. Les techniques de diagnostic à base de modèles sont les plus souples à ce sujet. Grâce aux normes de protocoles des réseaux et de gestion de réseaux, il est plus aisé d'établir un modèle de comportement du système en cas de pannes. En effet, certains mécanismes liés à la gestion de pannes sont décrits dans ces normes.

Les systèmes étudiés ont la particularité d'être de grande taille. Aussi les approches centralisées ne sont pas réalisables en pratique (le modèle global est trop important). Une approche décentralisée est bien adaptée, car elle est proche de la topologie du système qui est lui-même de nature décentralisée. Le défi consiste donc à mettre en place une architecture décentralisée pour le suivi d'un système complexe et le diagnostic en ligne informant sur les pannes et leur propagation dans le système. Cette architecture doit répondre à deux critères :

1. *efficacité :* elle doit être en mesure d'adapter son diagnostic au fur et à mesure que les alarmes sont reçues ;
2. *synthèse :* le diagnostic doit être le plus complet possible et présenté à l'Opérateur de la

façon la plus synthétique possible.

Les chapitres suivants décrivent une architecture de diagnostic qui essaie de répondre au mieux à ces critères. Dans le chapitre 3, nous présentons le cadre théorique de cette architecture ainsi que les difficultés liées à l'approche décentralisée. Les chapitre 4 et 5 décrivent la mise en œuvre de l'architecture décentralisée.

CHAPITRE 3

Diagnostic décentralisé : concepts et difficultés

Dans ce chapitre, nous introduisons la notion de diagnostic que nous utilisons dans notre approche. Cette notion se fonde sur un modèle de comportement du système, ce modèle étant représenté à l'aide d'un formalisme d'automates. Dans une seconde partie, nous présentons les difficultés liées à cette définition du diagnostic qui nous amènent à la mise en place d'une approche décentralisée pour la construction de ce diagnostic.

3.1 Exemple d'application

Dans cette section, un exemple d'application simple est présenté. Cet exemple d'étude va servir d'illustration aux différents concepts décrits par la suite. Cet exemple, que nous nommerons par la suite Toynet, est inspiré d'une application réelle : le réseau Transpac (voir section 6.3). Il s'agit d'un réseau constitué de trois commutateurs (*CM1*, *CM2*, *CM3*). Ces commutateurs ont pour objectif de faire transiter les informations sur un anneau (voir figure 3.1).

Chaque commutateur *CMi* est contrôlé par une station de contrôle *SCi*. Un centre de supervision *CS* a la charge de superviser les six équipements. Ce centre reçoit les alarmes venant des trois commutateurs *via* un réseau de gestion et des capteurs considérés comme fiables (pas de perte d'alarmes entre les commutateurs et le centre de supervision).

Un commutateur fonctionne de la façon suivante. Son comportement normal consiste à transmettre les paquets de données sur le réseau, il dispose pour cela de deux connexions : une connexion *ouest* (pour *CM1*, il s'agit de *cnx12*) et une connexion *est* (pour *CM1*, il s'agit de *cnx31*). Si la connexion est rompue, il émet une alarme (pour *CM1*, $CM1cx12$ pour *cnx12* et $CM1cx31$ pour *cnx31*) et il passe en mode d'attente. Si la connexion est rétablie, il reprend son fonctionnement normal. Un commutateur peut se bloquer, dans ce cas, un mécanisme d'alarmes informe le superviseur que le commutateur est bloqué (alarme *CMiblc*). La station de contrôle détecte également ce blocage et tente une action afin de réinitialiser le commutateur associé. Après la réinitialisation, le commutateur indique qu'il est opérationnel (alarme *CMiop*).

Une station de contrôle est sujette à deux types de pannes. Premièrement, elle peut se bloquer et reprendre son fonctionnement normal. Après un blocage, elle signale le fait qu'elle redevient opérationelle à l'aide d'une alarme *SCiop*. Cette alarme transite par le commutateur associé avant d'être envoyée au centre de supervision. Dans le cas où le commutateur est bloqué ou est en cours de réinitialisation, l'alarme émise par la station est masquée. La station

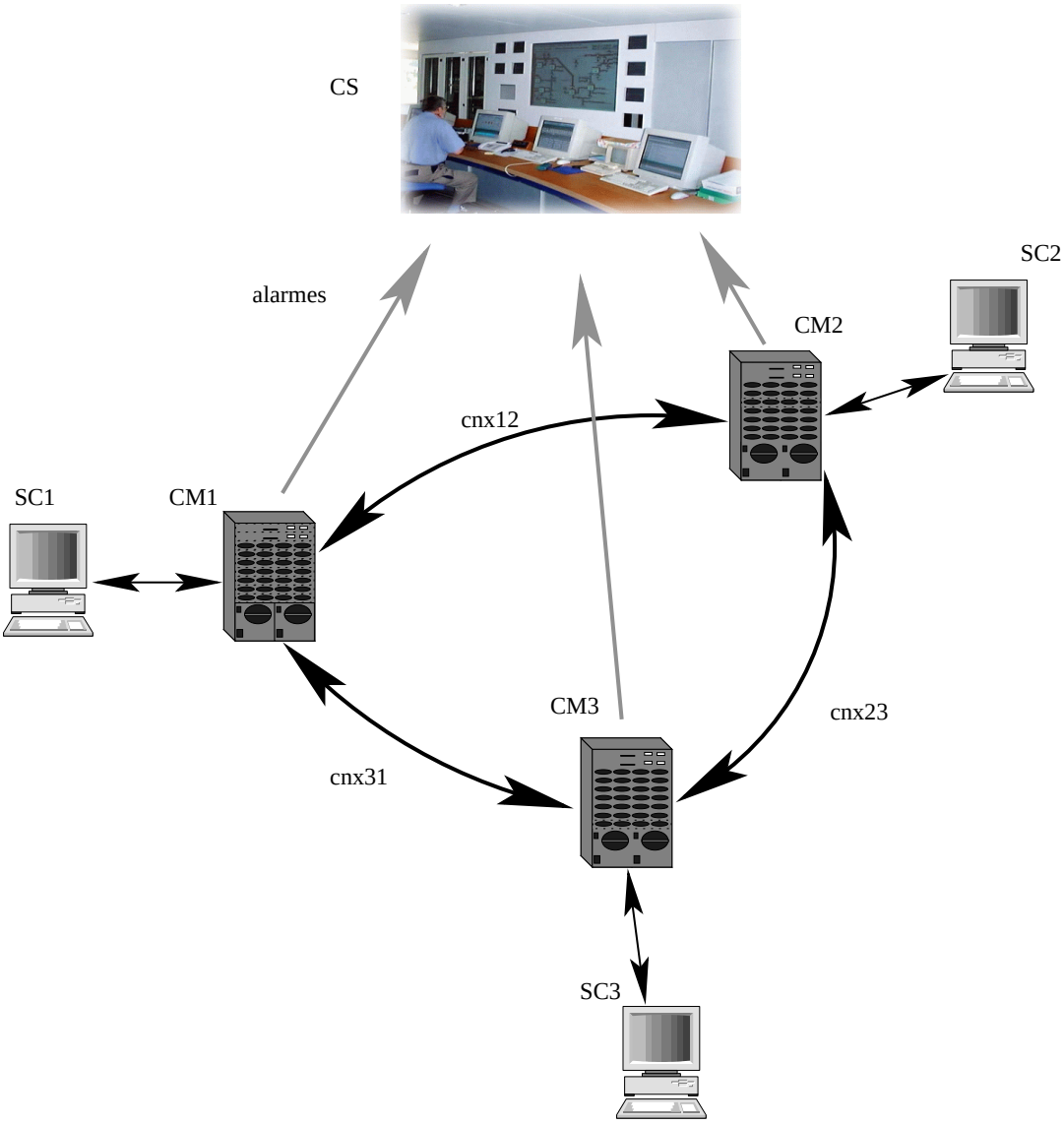

FIG. 3.1 – Topologie du réseau Toynet.

peut être réinitialisée (ou se réinitialiser spontanément). Une fois que la station a terminé sa réinitialisation, elle émet une alarme *SCiop* indiquant qu'elle est de nouveau opérationnelle, comme dans le cas où elle était bloquée.

3.2 Modèle

Cette section est consacrée à la description du formalisme que nous allons utiliser afin de modéliser des systèmes supervisés tels que celui présenté dans la section précédente.

3.2.1 Système et modèle

Un système est une entité situé dans un environnement. En général, un système n'est pas indépendant de son environnement, en ce sens où le système et son environnement sont en interactions (voir figure 3.2).

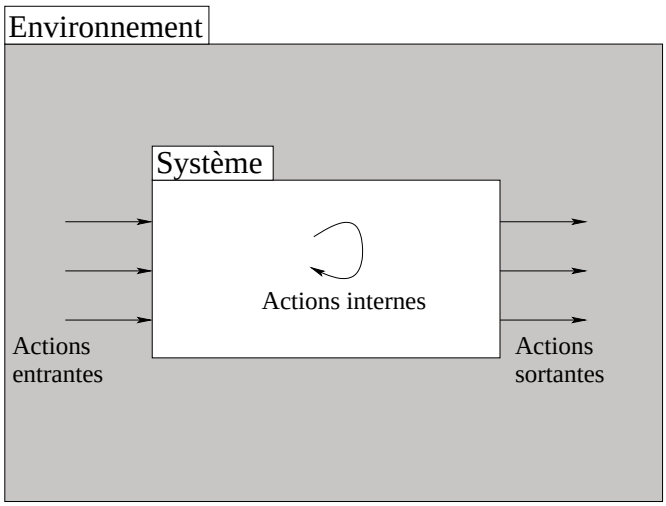

FIG. 3.2 – Interactions entre un système et son environnement.

Les interactions entre un système et son environnement peuvent être de diverses natures. L'environnement peut manipuler explicitement le système (un opérateur qui active des commandes du système) mais cette interaction peut être plus implicite (par exemple, le fonctionnement du système peut être sensible à la température de l'environnement, si cette température varie, l'environnement « agit » sur le système). De même, le système peut produire des actions sur l'environnement, comme dans le cas précédent, il peut s'agir de l'activation de commandes (si l'environnement en dispose) ou bien encore d'actions implicites (par exemple, si l'environnement dispose de capteurs observant le système, le système « agit » sur les capteurs si son comportement observable est modifié).

Établir un modèle du système pour en effectuer le diagnostic revient à mettre en relation ce que l'environnement est en mesure d'observer sur le système (un sous-ensemble des actions sortantes) et le fonctionnement normal et anormal du système. Ce fonctionnement est lié à

l'environnement : il dépend des actions de l'environnement pouvant le pertuber, en particulier des pannes. C'est dans cette optique qu'un modèle du système pour le diagnostic doit être établi. En particulier, ce modèle doit être une bonne abstraction du système dans le sens où il ne prend en compte que les informations nécessaires à la tâche de diagnostic souhaitée (par exemple, il est inutile de modéliser des comportements de système dont l'environnement n'est pas en mesure d'en observer une quelconque trace).

Les systèmes, tels que les réseaux de télécommunications, sont des ensembles de composants intégrés dans un environnement qui peut être vaste. Cet environnement est constitué entre autres d'un ou de plusieurs centres de supervisions qui « observent » le système avec des opérateurs qui peuvent contrôler le système. Mais cet environnement est constitué aussi des lieux où sont implantés les divers équipements (bâtiments, les endroits où passent les câbles du réseau...). La nature des systèmes considérés est discrète. Les composants du système interagissent entre eux essentiellement par des échanges de messages suivant des protocoles bien définis. De même, les interactions entre le système et son environnement sont de nature discrète, en particulier les observations sont des alarmes envoyées sous forme de notifications (voir section 1.4.2.2), les moyens de contrôle du système sont des commandes actionnées ou non. Ce fait nous incite donc à établir un modèle représentant un *système à événements discrets*. Il existe de nombreux travaux sur la modélisation de ce genre de système pour en effectuer le diagnostic, notamment par ordre chronologique [Riese 93c], [Bouloutas et al. 94], [Bibas et al. 96], [Boubour 97], [Sampath et al. 98], [Aghasaryan et al. 98], [Larsson 99], [Baroni et al. 00], [Rozé et Cordier 02]. Dans cette section, nous présentons un formalisme de représentation d'un système à événements discrets. Ce formalisme est utilisé pour établir le modèle abstrait du système supervisé sur lequel est fondée l'approche proposée. Ce formalisme est similaire à quelques détails près au formalisme décrit dans [Baroni et al. 00].

3.2.2 Système à événements discrets

3.2.2.1 Notion d'événements

La notion d'événements se produisant sur un système est très intuitive, si bien que nous n'allons pas chercher à en donner une définition formelle générale. Dans les systèmes que nous considérons, les événements peuvent représenter :

- des *actions* telles que « Un opérateur O appuie sur le bouton Y du composant A », « Activation de la réinitialisation du composant B » ;
- des *émissions* ou des *réceptions de messages* telles que « Envoi du message m du composant A vers le composant B », « Emission d'une alarme du type a par le composant A » ;
- des *propriétés* telles que « Le composant A commence à masquer le composant B », « Port d'entrée du composant A isolé ».

Une caractéristique importante d'un événement est son instantanéité, un événement n'a pas de durée. Tout au long de ce mémoire, nous considérerons que tout événement a une *origine* et une *cible*. L'origine d'un événement est l'entité qui a produit l'événement et la cible est l'entité qui le subit. Par exemple, pour l'événement « Un opérateur O appuie sur le bouton Y du

composant A », l'origine de l'événement est « l'opérateur O », autrement dit l'environnement du système, et la cible est « le composant A » (où le bouton Y est activé, ce qui peut modifier le comportement du composant A). Autre exemple, lorsque l'événement représente l'envoi d'un message, l'origine est le composant qui envoie le message et la cible est la connexion qui sert de support de communication pour ce message, de même, pour une réception, l'origine est la connexion et la cible le composant qui reçoit le message.

L'origine d'un événement est de deux types :

1. l'origine de l'événement est l'environnement : du point de vue du système, cet événement est *exogène* ;
2. l'origine de l'événement est une entité du système : cet événement est *endogène*.

3.2.2.2 Notion d'événement de pannes

Si nous reprenons la définition de panne (définition 1.1) telle que nous l'avons donnée dans la section 1.4.1, une panne est considérée comme un *état* de dysfonctionnement.

Du fait de la nature discrète des systèmes que nous cherchons à superviser, l'apparition d'une panne correspond à un changement d'état du système de même que sa disparition. Ainsi, l'occurrence d'une panne peut être caractérisée par l'occurrence d'un événement de *début de panne* et des événements de *retour en fonctionnement*.

Toute panne est ainsi représentée à l'aide d'événements. Une panne *permanente* est modélisée en particulier par l'occurrence d'un événement de début de panne et aucune occurrence d'événements de *retour en fonctionnement* à l'opposé de l'occurrence d'une panne *intermittente* qui est caractérisée par la présence d'événements de retour en fonctionnement.

Exemple Dans Toynet, le blocage d'un commutateur est considéré comme une panne. Cette panne est intermittente, car le commutateur peut être débloqué par une réinitialisation de la station de contrôle. L'occurrence d'un blocage sur le commutateur 1 peut être représentée par l'événement *CM1bloque*. La fin du bloquage se produit lorsque le commutateur redevient opérationnel après sa réinitialisation, cette fin est caractérisée par l'événement de retour en fonctionnement *CM1fin_réinit*. Dans Toynet, toutes les pannes sont considérées comme intermittentes. Le tableau 3.1 présente l'ensemble des événements de pannes et de retour en fonctionnement du réseau Toynet.

L'occurrence d'une panne est *primaire* (voir section 1.4.1) si l'occurrence de l'événement de début de panne est spontanée et n'est pas la conséquence d'un autre événement ayant pu avoir lieu dans le système. Aussi l'activation d'une panne primaire sera représentée par un événement exogène. Une panne primaire peut être vue comme une action de l'environnement sur le système lorsque certaines conditions sont réunies. Par exemple, la rupture d'une connexion physique entre deux composants est liée à une action de l'environnement sur cette connexion (« débranchement d'un câble »), le mauvais fonctionnement d'un équipement peut être provoqué par son usure au cours du temps (le temps faisant parti de l'environnement dans lequel évolue le système)...

Composant	Pannes primaires	Pannes secondaires	Retours
CM1	*CM1bloque*		*CM1fin_réinit*
		CM1attenteCnx12	*CM1finattenteCnx12*
		CM1attenteCnx31	*CM1finattenteCnx31*
SC1	*SC1bloque*		*SC1débloque*
	SC1réinit		*SC1fin_réinit*
cnx12	*ruptureCnx12*		*rétablissementCnx12*
CM2	*CM2bloque*		*CM2fin_réinit*
		CM2attenteCnx23	*CM2finattenteCnx23*
		CM2attenteCnx12	*CM2finattenteCnx12*
SC2	*SC2bloque*		*SC2débloque*
	SC2réinit		*SC2fin_réinit*
cnx23	*ruptureCnx23*		*rétablissementCnx23*
CM3	*CM3bloque*		*CM3fin_réinit*
		CM3attenteCnx23	*CM3finattenteCnx23*
		CM3attenteCnx31	*CM3finattenteCnx31*
SC3	*SC3bloque*		*SC3débloque*
	SC3réinit		*SC3fin_réinit*
cnx31	*ruptureCnx31*		*rétablissementCnx31*

TAB. 3.1 – Ensemble des événements de pannes et de retour en fonctionnement de Toynet.

Par opposition, l'occurrence de la panne est *secondaire* si l'événement qui produit la panne est le résultat de conditions internes au système provoquées par l'occurrence de pannes primaires. L'occurrence d'une panne secondaire sera donc représentée par l'occurrence d'un événement endogène du système, événement conséquence d'un événement exogène (autrement dit, conséquence d'une panne primaire).

Par définition, toute panne secondaire est la conséquence d'au moins une panne primaire. Cette conséquence peut être directe, la panne primaire provoque la panne secondaire, ou indirecte, la panne secondaire est provoquée par une chaîne de pannes secondaires issues d'au moins une panne primaire : il s'agit de la *propagation des pannes*. Le modèle du système représente donc toutes les propagations de pannes primaires possibles en se fondant sur le comportement du système et sur un ensemble de pannes recensées.

Exemple La rupture de la connexion *cnx12* représentée par l'occurrence de l'événement *ruptureCnx12* est considérée comme spontanée : il s'agit d'une panne primaire. Par contre, cette panne provoque aussi la mise en attente des commutateurs (événements *CM1attenteCnx12* et *CM2attenteCnx12*) en bout de cette connexion : ces mises en attente sont des pannes secondaires (voir figure 3.3). L'apparition d'un événement de retour en fonctionnement de la connexion au bout d'un certain temps produit la fin de l'attente des deux commutateurs.

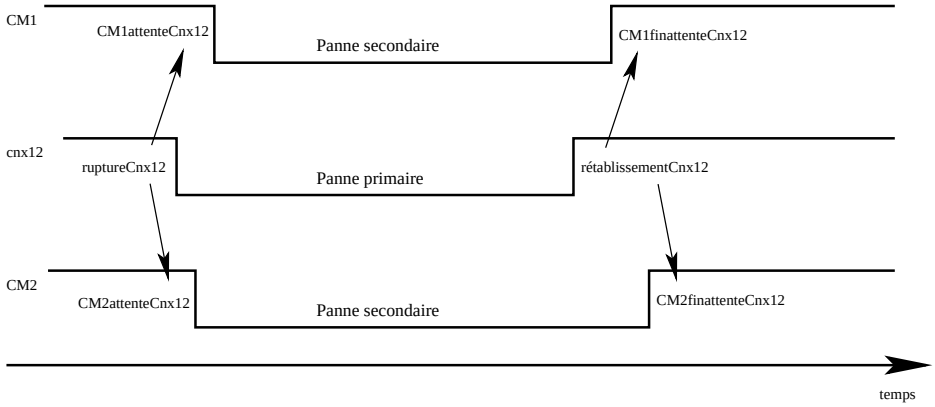

FIG. 3.3 – Propagation de pannes.

3.2.3 Modèle décentralisé

Un système est constitué d'un ensemble d'entités ou composants (physiques ou logiques) qui sont plus ou moins complexes. Une des premières tâches à effectuer lors de la phase de modélisation d'un tel système est de trouver le niveau d'abstraction qui facilite la modélisation du système tout en conservant toutes les informations nécessaires (la granularité) à l'exploitation du modèle. Dans une approche décentralisée, ce niveau d'abstraction est caractérisé par la notion de *composant élémentaire*. Le modèle décentralisé exprime le comportement de chacun des composants élémentaires ainsi que la façon dont ils communiquent avec l'environnement ou avec les autres composants.

3.2.3.1 Composant élémentaire

Un composant élémentaire est une partie du système choisie selon des critères liés à la modélisation. En tout premier lieu, le comportement de ce composant est pertinent dans le sens où il peut tomber en panne ou servir de support à la propagation de pannes dans le système. Un composant élémentaire doit être simple à modéliser dans le sens où cela doit être naturel : il peut s'agir d'un composant (physique ou logique) complet du système ou d'une partie parfaitement délimitée de ce composant, d'un groupe de composants. Le comportement du composant élémentaire n'est pas décomposable ou alors cette décomposition n'est pas souhaitée, il constitue une « brique » du comportement du système. Dans le cadre de Toynet, nous allons considérer les types de composants élémentaires suivants :

1. la *station de contrôle* (*SC1*, *SC2*, *SC3*) ;
2. la partie *gestion des connexions* du commutateur qui représente le comportement du commutateur par rapport aux connexions adjacentes (*CM1cnx*, *CM2cnx*, *CM3cnx*) ;
3. la partie *contrôle* qui représente le comportement du commutateur par rapport à la station de contrôle (*CM1ctl*, *CM2ctl*, *CM3ctl*) ;
4. la *connexion* entre les commutateurs (*cnx12*, *cnx23*, *cnx31*).

Le choix de ces composants élémentaires respecte les critères définis auparavant. La station de contrôle représente un équipement du réseau qui est simple à modéliser, elle n'interagit qu'avec le commutateur associé. Le commutateur est un équipement plus complexe qui peut être divisé en deux parties. Cette division simplifie la modélisation. On considère également les connexions comme faisant partie des composants élémentaires. En effet, les connexions sont sujettes à des pannes et l'information nécessaire à discriminer ces pannes est disponible dans le comportement des commutateurs. Ainsi, Toynet est constitué de 12 composants élémentaires.

3.2.3.2 Communications entre entités du système

Les communications entre deux entités d'un système ou bien même entre le système et son environnement peuvent être de différentes natures. En particulier, dans le cadre des réseaux de télécommunications, ces communications sont généralement effectuées à l'aide de connexions sur lesquelles transitent des messages. Ces connexions font partie du comportement du système en cas de pannes et doivent donc être prises en compte. Tout au long de ce mémoire, nous considérerons que les connexions font partie intégrante du modèle de comportement du système et seront toujours considérées comme des composants élémentaires à part entière. Chaque connexion peut bien entendu avoir un comportement, suivre une politique qui lui est propre ([Baroni et al. 98] en cite certaines). En particulier, on peut modéliser une connexion par une file dans laquelle transitent des événements. L'ajout d'un tel composant élémentaire est nécessaire pour modéliser des échanges de messages dont le délai entre l'émission et la réception peut avoir un impact sur le comportement futur du système : ce cas se produit par exemple si le composant élémentaire en charge de la réception de l'événement peut en recevoir un autre alors que le premier est en transit ; suivant la réception ou non de cet autre événement, la réception de l'événement en transit peut conduire à des comportements différents. Une conséquence de cette modélisation est l'augmentation du nombre de comportements possibles du système ; il faut donc en user avec parcimonie. Néanmoins, dans tous les cas, nous ferons l'hypothèse suivante.

Hypothèse 3.1 (Connexion bornée) *Sur toute connexion d'une entité vers une autre ou vers l'environnement du système, le nombre de messages en transit est fini.* □

Dans Toynet, pour des raisons de simplicité, nous considérons que les communications entre les entités (stations de contrôle et commutateurs) sont instantanées.

3.2.3.3 Modèle d'un composant élémentaire

Un composant élémentaire réagit soit à des événements exogènes du système, soit à des événements internes produits par un autre composant élémentaire. Ce composant peut répondre à de tels événements en produisant des événements vers l'environnement ou vers d'autres composants élémentaires du système. Ainsi, le modèle d'un composant élémentaire doit exprimer la réponse à un *stimulus* (une séquence d'événements) par une autre séquence d'événements. Ce principe peut être facilement modélisé à l'aide d'un *transducteur* dont l'objectif est de modéliser la traduction d'une séquence d'entrée en une séquence de sortie [Aho et Ullman 72].

Définition 3.1 (Modèle d'un composant élémentaire) *Le modèle d'un composant élémentaire est un transducteur Γ_i :*

$$\Gamma_i = (\Sigma^i_{dec}, \Sigma^i_{émis}, Q_i, E_i)$$

- Σ^i_{dec} *est l'ensemble des événements déclencheurs (événements dont la cible est Γ_i) ;*
- $\Sigma^i_{émis}$ *est l'ensemble des événements émis par le composant (événements dont l'origine est Γ_i) ;*
- $\Sigma^i_{dec} \cap \Sigma^i_{émis} = \emptyset$ *;*
- Q_i *est l'ensemble des états du composant ;*
- $E_i \subseteq (Q_i \times \Sigma^i_{dec} \times 2^{(\Sigma^i_{émis})} \times Q_i)$ *est l'ensemble des transitions.*

□

Une transition $(q, dec, \mathcal{E}, q')$ du modèle représente la réaction du composant quand celui-ci est dans l'état q et qu'il reçoit l'événement dec (l'événement dec a donc pour cible ce composant élémentaire), à savoir l'émission instantanée de l'ensemble d'événements $\mathcal{E}$ et le passage dans l'état q' (tout événement de $\mathcal{E}$ a donc pour origine ce composant élémentaire). Une telle transition sera notée $q \xrightarrow{dec|\mathcal{E}} q'$. On considère dans ce modèle que le composant ne réagit pas à un événement qu'il a lui-même provoqué (pas de rétro-réaction, l'origine d'un événement est toujours différente de sa cible).

Le modèle Γ_i peut être non-déterministe sur les événements déclencheurs, c-à-d qu'il peut posséder deux transitions $q \xrightarrow{dec|\mathcal{E}} q'$ et $q \xrightarrow{dec|\mathcal{E}'} q''$ où $\mathcal{E} \neq \mathcal{E}'$ et où q' et q'' peuvent être différents ou non. Ce non-déterminisme offre la possibilité de modéliser différentes hypothèses de comportement d'un composant face à l'occurrence d'une panne (comportement non-déterministe).

Sur la figure 3.4, le transducteur associé à la partie *contrôle* du commutateur *CM1* est présenté (les autres sont également présentés dans l'annexe A). L'ensemble des événements déclencheurs est

$$\Sigma^{CM1_{ctl}}_{dec} = \{\textit{SC1opérationnel}, \textit{CM1bloque}, \textit{chgCx12CM1}, \textit{chgCx31CM1},$$
$$\textit{CM1réinit}, \textit{CM1fin\_réinit}\}$$

et l'ensemble des événements émis est

$$\Sigma^{CM1_{ctl}}_{émis} = \{\textit{CM1a\_relancer}, \textit{CM1blc}, \textit{CM1op}, \textit{CM1cx31}, \textit{CM1cx12}, \textit{SC1op}\}.$$

L'ensemble des états est $Q_{CM1_{ctl}} = \{e_1, e_2, e_3\}$ correspondant respectivement à l'état nominal de $CM1_{ctl}$, l'état de blocage et l'état de réinitialisation.

SC1opérationnel est un événement dont l'origine est *SC1* et la cible est *CM1ctl*. Il indique à la partie contrôle du commutateur que la station de contrôle est opérationnelle. Si le commutateur fonctionne correctement, il émet l'alarme *SC1op*, sinon il n'émet rien. *CM1bloque* est un événement de début de la panne *blocage de CM1*. La station de contrôle en est informée par l'émission de l'événement *CM1a_relancer* de même que le superviseur par l'émission de l'alarme *CM1blc*. L'événement *CM1réinit* est envoyé par la station de contrôle et modélise la réinitialisation du commutateur, quant à *CM1fin_réinit*, il modélise la fin de cette

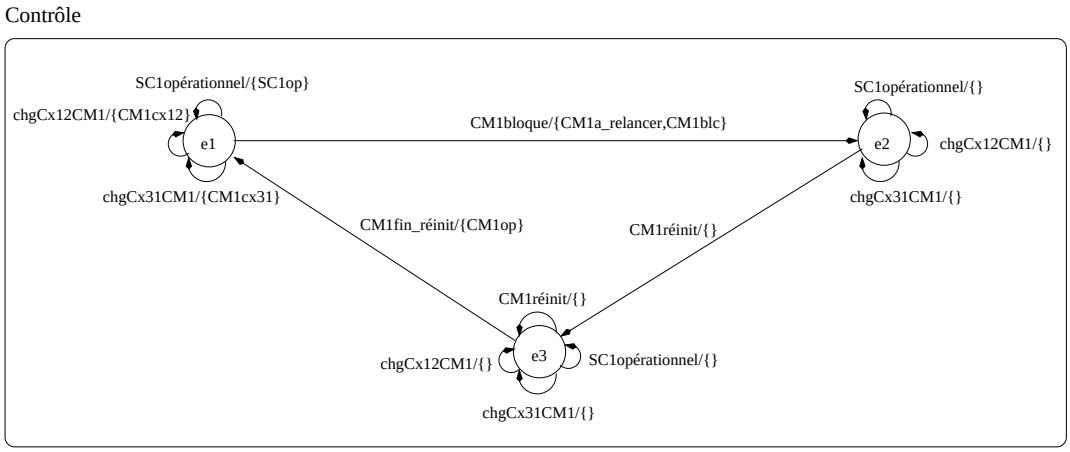

FIG. 3.4 – Composant élémentaire représentant la partie *contrôle* de l'équipement *CM1*.

réinitialisation. Les événements *chgCx12CM1* et *chgCx31CM1* sont envoyés par la partie *gestion des connexions* du commutateur et indiquent un changement de statut des connexions. En fonction de l'état du commutateur, les alarmes correspondantes *CM1cx12* ou *CM1cx31* sont ou non émises.

3.2.3.4 Définition du modèle décentralisé

Le *modèle décentralisé* du système est donc un ensemble de transducteurs (on parle aussi d'automates communicants), chaque transducteur représentant un composant élémentaire.

Définition 3.2 (Modèle décentralisé) *Le* modèle décentralisé *du système est un ensemble de modèles de composants élémentaires* $\Gamma \triangleq \{\Gamma_1, \ldots, \Gamma_n\}$ *tel que :*

1. $\forall i, j \in \{1, \ldots, n\}, i \neq j, \Sigma^i_{dec} \cap \Sigma^j_{dec} = \emptyset$;
2. $\forall i, j \in \{1, \ldots, n\}, i \neq j, \Sigma^i_{émis} \cap \Sigma^j_{émis} = \emptyset$.

□

Les 2 conditions sur les événements définissent la partie structurelle du modèle décentralisé. La condition 1 assure qu'un événement ne peut atteindre qu'un seul composant élémentaire (un événement n'a qu'une seule cible). De même, la condition 2 assure qu'un événement ne peut être provoqué que par un seul composant (un événement n'a qu'une seule origine). Ces 2 conditions supposent que tout événement est localisé (on connaît son origine et sa cible).

Les différentes notions d'événements qui ont été vues auparavant sont caractérisées formellement à partir du modèle Γ.

Définition 3.3 (événement exogène) *L'ensemble des* événements exogènes *de* Γ *est l'ensemble des événements :*

$$\Sigma_{exo} \triangleq \bigcup_{i=1}^{n} \Sigma^i_{dec} \setminus \bigcup_{i=1}^{n} \Sigma^i_{émis}.$$

□

Cette définition exprime le fait qu'un événement exogène du système est un événement qui déclenche une réaction sur un composant élémentaire et qui n'est pas le produit d'un autre composant élémentaire. Par la suite, nous considérerons que tout événement exogène est un événement qui active ou annihile une panne ou plus généralement un état que l'on veut pouvoir diagnostiquer. Tout autre événement exogène n'a pas d'intérêt si l'objectif final n'est pas d'en discerner la présence.

Définition 3.4 (événement endogène) *L'ensemble des* événements endogènes *de* Γ *est constitué de deux sous-ensembles distincts* $\Sigma_{endo} = \Sigma_{prod} \cup \Sigma_{int}$ *tels que :*

1. Σ_{prod} *(*événements produits*) est l'ensemble des événements produits par le système et émis vers son environnement :*

$$\Sigma_{prod} \triangleq \bigcup_{i=1}^{n} \Sigma^i_{émis} \setminus \bigcup_{i=1}^{n} \Sigma^i_{dec}$$

2. Σ_{int} *(*événements internes*) est l'ensemble des événements émis et reçus par les composants élémentaires du système :*

$$\Sigma_{int} \triangleq \bigcup_{i=1}^{n} \Sigma^i_{émis} \cap \bigcup_{i=1}^{n} \Sigma^i_{dec}.$$

□

La distinction des événements endogènes en 2 sous-ensembles est importante du point du vue du diagnostic. Dans le cadre d'un système muni d'un superviseur (superviseur appartenant à l'environnement du système), on considère en effet que seuls les événements produits par le système vers son environnement peuvent être *observables* : dans le cadre des réseaux de télécommunications, il s'agit de la réception d'une alarme. Quant aux *événements internes*, ils sont liés au comportement du système et participent à la propagation des pannes, ils ont la caractéristique d'être *non observables* (un événement interne est l'échange d'une information entre deux composants élémentaires). Dans la suite de ce mémoire, nous considérerons que tout événement produit par le système vers son environnement est observable. Si l'on note par Σ_{obs} l'ensemble des événements observables du système, on considèrera par la suite que :

$$\Sigma_{obs} = \Sigma_{prod}.$$

Sur la figure 3.5, la partie du modèle structurel concernant le commutateur *CM1* est présentée. Chaque flèche correspond à un sens de propagation entre deux composants élémentaires ou entre un composant élémentaire et l'environnement. L'étiquette associée à chaque flèche contient l'ensemble des événements contribuant à l'interaction. Les événements internes sont ceux appartenant à des étiquettes de flèches joignant deux composants élémentaires. L'ensemble des événements produits le sont par la partie contrôle du commutateur. Les différents composants élémentaires sont sujets à des pannes primaires (événements exogènes).

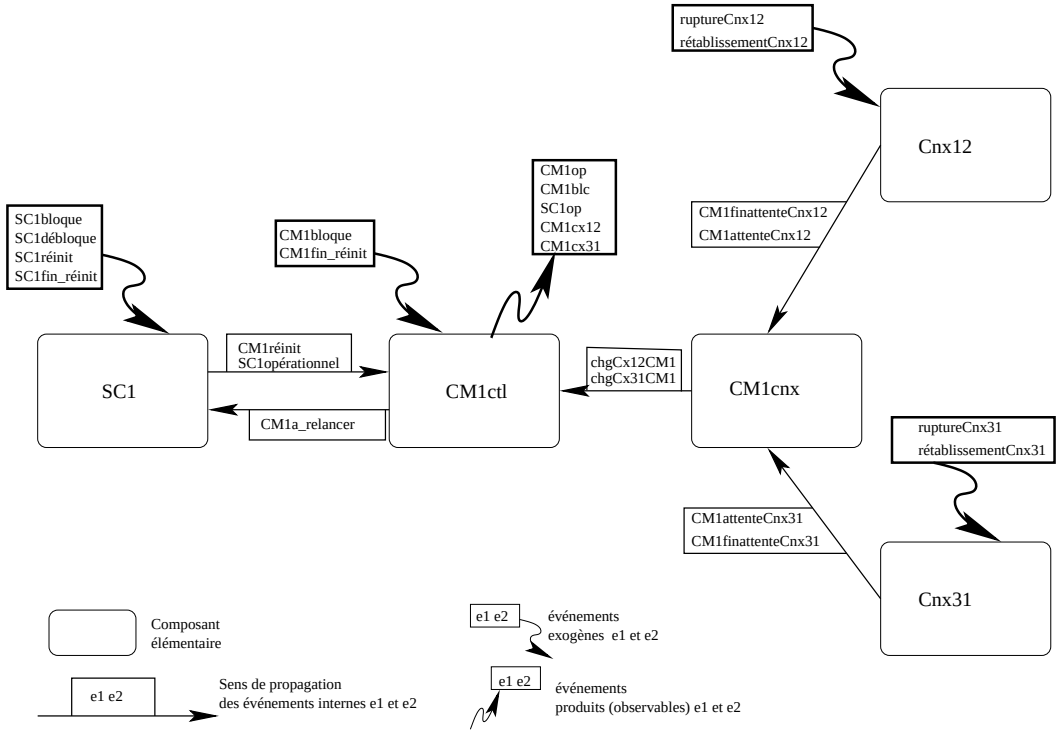

FIG. 3.5 – Partie du modèle structurel de Toynet (voisinage de *CM1*).

3.2.4 Sémantique du modèle décentralisé

Dans cette section est formellement décrite la sémantique d'un tel modèle. Le comportement du système (comportement global) est défini à partir des modèles de composants élémentaires et à l'aide d'une opération sur ces modèles : le produit synchronisé de systèmes de transitions ([Arnold et Nivat 82], [Arnold 92]).

3.2.4.1 Produit libre de systèmes de transitions

Le *produit libre* est une opération sur des systèmes de transitions. Elle est présentée ici dans le cadre des transducteurs.

Définition 3.5 (produit libre) *Le* produit libre *de m transducteurs $T_i = (I_i, O_i, Q_i, E_i), i \in \{1, \ldots, m\}$ est le transducteur (I, O, Q, E) tel que :*
- *$I = I_1 \times \cdots \times I_m$;*
- *$O = O_1 \times \cdots \times O_m$;*
- *$Q = Q_1 \times \cdots \times Q_m$ est l'ensemble des états ;*
- *$E = E_1 \times \cdots \times E_m$ est l'ensemble des transitions*

$$(q_1, \ldots, q_m) \overset{(t_1,\ldots,t_m)}{\longrightarrow} (q'_1, \ldots, q'_m) = (q_1 \overset{t_1}{\longrightarrow} q'_1, \ldots, q_m \overset{t_m}{\longrightarrow} q'_m).$$

□

Par la suite, nous noterons ce produit par $\langle T_1, \ldots, T_m \rangle$. Par définition de ce produit, il est facile de voir que le transducteur $\langle T_{j_1}, \ldots, T_{j_m} \rangle$ est isomorphe au transducteur $\langle T_1, \ldots, T_m \rangle$ pour toute permutation $\{j_1, \ldots, j_m\}$ de $\{1, \ldots, m\}$.

La sémantique du modèle décentralisé s'appuie sur un tel produit.

3.2.4.2 Hypothèses sur l'activation de transitions

Du fait de la nature instantanée des événements, on considérera l'hypothèse suivante.

Hypothèse 3.2 *Deux événements exogènes du système ne peuvent être reçus en même temps.* □

Les événements exogènes se succèdent et provoquent le passage d'un état à un autre selon les transitions étiquetées par ces événements. Lors de la transition, d'autres événements sont émis et activent d'autres transitions du modèle décentralisé ; ainsi est exprimée la propagation des pannes dans le système. Cette activation de transitions est sujette à une deuxième hypothèse.

Hypothèse 3.3 *Toute propagation instantanée d'un événement exogène sur le système est acyclique.* □

Cette hypothèse signifie que toute propagation d'un événement exogène parmi les composants élémentaires a une forme d'arbre (voir figure 3.6). La racine de l'arbre est le composant

affecté directement par l'événement exogène. Cet événement active une transition du composant qui produit à son tour des événements. Les fils d'un nœud sont les composants (ou les connexions) affectés par les événements produits par la transition activée dans le nœud père.

Sémantiquement, cette hypothèse exprime le fait que deux événements internes ne peuvent se produire en même temps sur un même composant élémentaire : du fait de l'instantanéïté des évenements, deux événements se produisent toujours l'un après l'autre. Cette hypothèse n'interdit pas la modélisation de phénomènes de rétro-propagation mais justifie le fait qu'un phénomène de rétro-propagation ne peut pas être instantané.

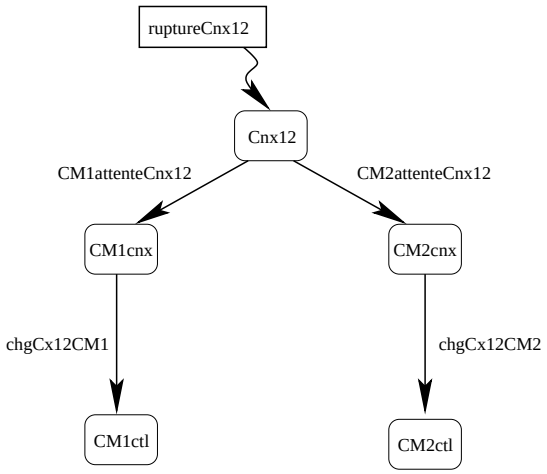

FIG. 3.6 – Hypothèse : propagation instantanée de pannes acyclique.

3.2.4.3 Synchronisation

Le modèle décentralisé dispose d'une *interprétation asynchrone*. Pour des raisons de simplicité d'écriture, le modèle est adapté afin d'en avoir une *interprétation synchrone* équivalente. Pour cela, des transitions nulles (notées $q \xrightarrow{\mathbf{e}|\{\}} q'$) sont ajoutées à chaque état de chaque tranducteur. Une telle transition d'un état sur lui-même signifie qu'un composant peut rester dans un état alors que les autres peuvent évoluer. Ces transitions ont pour but d'exprimer le fait qu'un composant peut rester inactif alors qu'un autre est activé. Cette adaptation permet d'interpréter l'ensemble des modèles de façon synchrone (à savoir que tout composant dispose d'une transition active en même temps que les autres). L'ensemble des transitions nulles de Γ_i sera noté N_i. Grâce à cette adaptation, l'ensemble des comportements possibles du système est inclus dans le produit libre :

$$\langle \Gamma_1, \ldots, \Gamma_n \rangle = (I, O, Q, E)$$

où
- $I = (\Sigma^1_{dec} \cup \{\mathbf{e}\}) \times \cdots \times (\Sigma^n_{dec} \cup \{\mathbf{e}\})$ est l'ensemble des événements déclencheurs ;
- $O = \Sigma^1_{émis} \times \cdots \times \Sigma^n_{émis}$ est l'ensemble des événements émis ;
- $Q = Q_1 \times \cdots \times Q_n$ est l'ensemble des états ;

– $E = (E_1 \cup N_1) \times \cdots \times (E_n \cup N_n)$ est l'ensemble des transitions.

Définition 3.6 (transition synchronisée) *On dit qu'une transition* $x \xrightarrow{t} x'$ *du produit de* $\langle \Gamma_1, \ldots, \Gamma_n \rangle$ *est* synchronisée *si et seulement si :*

– $q \xrightarrow{t} q' = (q_1 \xrightarrow{t_1} q'_1, \ldots, q_n \xrightarrow{t_n} q'_n)$ *;*
– $\exists! j \in \{1, \ldots, n\} | (t_j = e_j | \mathcal{E}_j) \wedge e_j \in \Sigma_{exo}$ *;*
– *pour chaque* j *de* $\{1, \ldots, n\}$ *tel que* t_j *est non nulle, on a* $t_j = e_j | \mathcal{E}_j \wedge \forall e \in \mathcal{E}_j \cap I, \exists l \in \{1, \ldots, n\}, t_l = e | \mathcal{E}_l$.

□

Une transition synchronisée $t = (t_1, \ldots, t_n)$ signifie que toutes les transitions t_i émettant des événements internes à $\{\Gamma_1, \ldots, \Gamma_n\}$ sont synchronisées avec des transitions t_j déclenchées par ces événements émis. De plus, elle contient une transition unique déclenchée par un événement exogène. Ainsi, une transition synchronisée regroupe les transitions des composants élémentaires activées lors de la propagation liée à l'événement exogène en question. L'ensemble des transitions synchronisées du produit libre $\langle \Gamma_1, \ldots, \Gamma_n \rangle$ définit des contraintes de synchronisation.

Notations : les transitions synchronisées sont de la forme :

$$(q_1 \xrightarrow{e_1 | \mathcal{I}_1 \cup \mathcal{O}_1} q'_1, \ldots, q_n \xrightarrow{e_n | \mathcal{I}_n \cup \mathcal{O}_n} q'_n).$$

$\mathcal{I}_i$ et $\mathcal{O}_i$ sont respectivement l'ensemble des événements internes et l'ensemble des événements produits associés à la transition $q_i \xrightarrow{e_i | \mathcal{I}_i \cup \mathcal{O}_i} q'_i$. Tout événement de $\mathcal{I}_i$ est un événement qui déclenche une transition $q_j \xrightarrow{e_j | \mathcal{I}_j \cup \mathcal{O}_j} q'_j, j \neq i$, de plus, dans l'ensemble des événements $\{e_1, \ldots, e_n\}$ il existe un unique événement e_{exo} exogène qui déclenche cette transition. Aussi, afin de simplifier la notation, une telle transition sera exprimée de la façon suivante :

$$(q_1, \ldots, q_n) \xrightarrow{e_{exo} | \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_n \; \mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n} (q'_1, \ldots, q'_n).$$

Exemple La figure 3.7 présente une transition synchronisée du produit des 12 composants élémentaires de Toynet. La figure présente les transitions non nulles concernées, les autres appartenant aux autres composants élémentaires étant du type $q_i \xrightarrow{\mathbf{e}} q_i$. Cette transition exprime la propagation d'un événement de panne exogène qui exprime la rupture de la connexion entre le commutateur 1 et le commutateur 2 (propagation présentée aussi sur la figure 3.6). Le gestionnaire de connexion de chaque commutateur s'en rend compte et informe la partie contrôle du commutateur. Le commutateur 1 est dans un mode où il est en mesure d'envoyer une alarme, ce qui n'est pas le cas du commutateur 2 (si ça avait été le cas, deux alarmes auraient été émises).

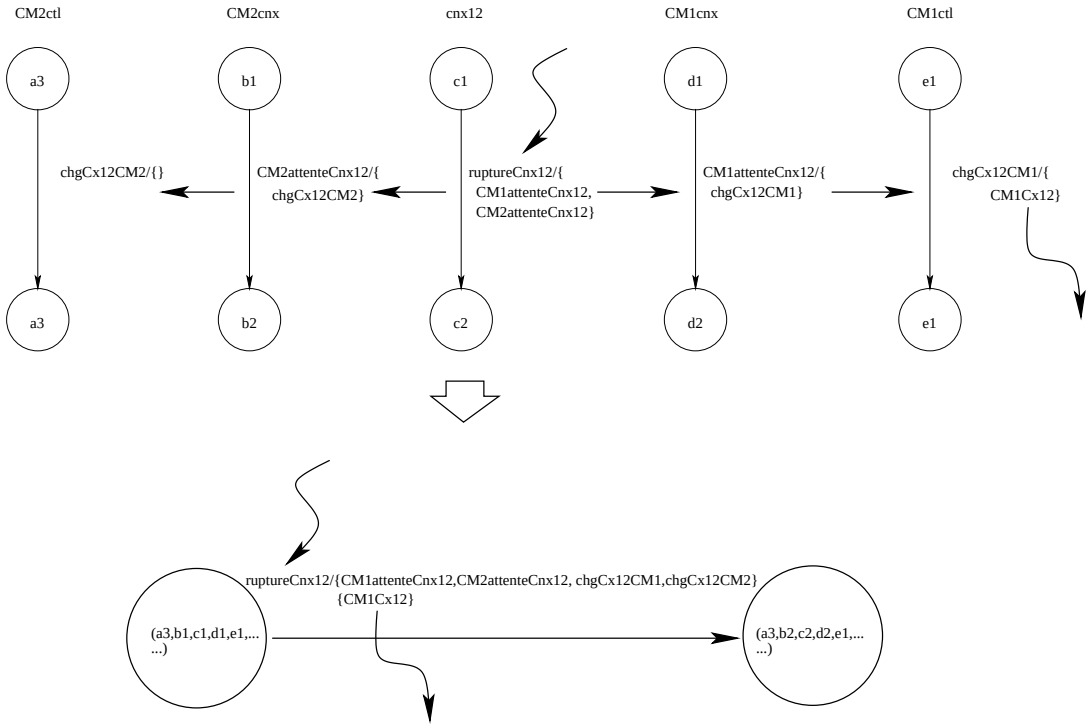

FIG. 3.7 – Transition synchronisée : propagation d'une rupture de la connexion *cnx12*.

3.2.4.4 Comportement global

Le *comportement global* associé au modèle $\Gamma = \{\Gamma_1, \ldots, \Gamma_n\}$ et noté $\|\Gamma\| = \|\Gamma_1, \ldots, \Gamma_n\|$ est défini par un sous-ensemble du produit libre respectant les contraintes de synchronisation.

Définition 3.7 (Comportement global) *Le* comportement global $\|\Gamma_1, \ldots, \Gamma_n\|$ *est le transducteur* (I, O, Q, E') *issu du produit libre* $\langle \Gamma_1, \ldots, \Gamma_n \rangle$ *tel que :*

– $E' \subseteq E$ *est l'ensemble des transitions synchronisées de* E *;*

□

Le comportement global $\|\Gamma\|$ représente la réaction instantanée du système à un stimulus externe (autrement dit, un événement exogène du système). Par définition, $\|\Gamma_1, \ldots, \Gamma_n\|$ est égal, à un isomporphisme près, à $\|\Gamma_{j_1}, \ldots, \Gamma_{j_n}\|$ où $j_1, j_2, \ldots, j_n$ est une permutation de $1, 2, \ldots, n$, autrement dit il n'y a qu'un seul comportement associé à l'ensemble $\{\Gamma_1, \ldots, \Gamma_n\}$.

En reprenant la forme simplifiée des transitions synchronisées, toute transition du comportement global est de la forme :

$$(q_1, \ldots, q_n) \xrightarrow{e_{exo} | \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_n \; \mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n} (q'_1, \ldots, q'_n).$$

Cette forme exprime bien le fait que si le système est dans l'état $(q_1, \ldots, q_n)$ et que l'événement e_{exo} se produit sur le système alors le système réagit en produisant des événements observables $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n$ et des événements internes $\mathcal{I}_1 \cup \cdots \cup \mathcal{I}_n$ qui le font aboutir dans l'état $(q'_1, \ldots, q'_n)$. Parmi les événements exogènes, il y a en particulier les événements de pannes (pannes primaires) et parmi les événements de $\mathcal{I}_1 \cup \cdots \cup \mathcal{I}_n$, il y a des événements de pannes s'étant produits par propagation de la panne primaire : ce sont les événements de pannes secondaires (voir figure 3.7).

3.3 Diagnostic du système

3.3.1 Caractérisation du diagnostic

Dans le chapitre 2, nous avons présenté plusieurs techniques afin d'élaborer un diagnostic. Nous avons vu en particulier qu'il en existait plusieurs types qui dépendent de l'information utilisée et disponible. Ces diagnostics vont de la détection de pannes (un composant est en panne), à la localisation de pannes (le composant i est en panne) jusqu'à l'identification de la panne (le composant i subit la panne p).

Dans le cadre de la supervision des réseaux de télécommunications, non seulement l'identification des pannes est nécessaire mais aussi la façon dont elles expliquent les alarmes reçues par le superviseur. Le diagnostic de réseau se doit d'être très riche : il doit permettre de localiser et d'identifier différents types de pannes et expliquer la propagation des alarmes liées à ces pannes. Le diagnostic doit aussi prendre en compte les pannes intermittentes afin d'expliquer le flot d'alarmes reçues. Son dernier objectif est aussi de proposer un ensemble d'états possibles du système à chaque instant en relation avec le flot d'alarmes déjà reçues.

Toutes les informations utiles au diagnostic sont contenues dans le modèle : événements de panne (intermittentes ou non), notion d'états du système, propagation des pannes dans le

système. C'est à partir de ce modèle que la définition du diagnostic est donnée. Cette définition nécessite au préalable l'introduction de différentes notions.

3.3.2 Notions sur les ensembles partiellement ordonnés

Définition 3.8 (ordre partiel) *Soit E un ensemble, un* ordre partiel *sur E est une relation $\preceq$ telle que pour tout $(x, y, z) \in E^3$:*

- $x \preceq x$ *(réflexivité) ;*
- $(x \preceq y \wedge y \preceq x) \Rightarrow x = y$ *(symétrie) ;*
- $(x \preceq y \wedge y \preceq z) \Rightarrow x \preceq z$ *(transitivité).*

L'ensemble E muni d'un tel ordre est appelé ensemble partiellement ordonné. □

Définition 3.9 (ensemble induit) *Soit $E = (\Sigma, \preceq)$ un ensemble partiellement ordonné constitué des éléments de Σ, soit $E_{ind} = (\Sigma_{ind}, \preceq_{ind})$ tel que $\Sigma_{ind} \subseteq \Sigma$ et $\forall e_1, e_2 \in \Sigma_{ind}, e_1 \preceq_{ind} e_2 \equiv e_1 \preceq e_2$, on dit que E_{ind} est un ensemble partiellement ordonné* induit *de E.* □

Définition 3.10 (séquence admissible) *Soit $E = (\Sigma, \preceq)$ un ensemble partiellement ordonné constitué des éléments de Σ, une* séquence admissible *σ de E est un ensemble totalement ordonné $(\Sigma, \preceq_\sigma)$. La relation d'ordre total $\preceq_\sigma$ doit vérifier : si $e_1 \preceq e_2$ alors $e_1 \preceq_\sigma e_2$.* □

Une séquence admissible σ de E est donc une séquence des éléments de E qui respecte l'ordre partiel associé à E.

Définition 3.11 (ensemble préfixe) *Soit $E_1 = (\Sigma_1, \preceq_1)$ et $E_2 = (\Sigma_2, \preceq_2)$ deux ensembles partiellement ordonnés, on dit que E_1 est un* ensemble préfixe *de E_2 (noté $E_1 \sqsubseteq E_2$), si toute séquence σ_1 admissible de E_1 est telle qu'il existe dans E_2, une séquence admissible $\sigma = \sigma_1\sigma_2$.* □

L'ensemble des ensembles préfixes de E sera noté $Pr(E)$. L'ensemble vide $\emptyset$ est préfixe de tout ensemble E : $\emptyset \in Pr(E)$. L'ensemble E est préfixe de lui-même : $E \in Pr(E)$. Tout ensemble préfixe de E est un ensemble induit de E.

Définition 3.12 (ensembles joints) *Soit $E_1 = (\Sigma, \preceq_1)$ et $E_2 = (\Sigma, \preceq_2)$ deux ensembles partiellement ordonnés, la jointure de E_1 et de E_2 est l'ensemble partiellement ordonné $E_1 \diamond E_2 = (\Sigma, \preceq_{12})$ tel que :*

$$\forall e_1, e_2 \in \Sigma, e_1 \preceq_{12} e_2 \equiv (e_1 \preceq_1 e_2 \vee e_1 \preceq_2 e_2$$

$$\vee(\exists e_3 \in \Sigma | e_1 \neq e_3 \neq e_2 \wedge ((e_1 \preceq_1 e_3 \wedge e_3 \preceq_2 e_2) \vee (e_1 \preceq_2 e_3 \wedge e_3 \preceq_1 e_2))$$

□

La jointure de deux ensembles E_1 et E_2 est l'ensemble des mêmes éléments que ceux de E_1 et de E_2 muni d'une relation d'ordre partiel plus stricte. La jointure n'est pas définie partout, il se peut que par construction de $\preceq_{12}$, cette relation ne soit plus une relation d'ordre. Par construction, $\preceq_{12}$ est nécessairement réflexive et transitive, par contre on ne garantit pas la symétrie de $\preceq_{12}$. Dans le cas de la non-symétrie de $\preceq_{12}$, on dira que la jointure n'existe pas, cela se produit lorsque les relations d'ordre $\preceq_1$ et $\preceq_2$ sont *incompatibles* entre elles, à savoir ssi :

$$\exists e_1, e_2 \in \Sigma | e_1 \neq e_2 \wedge e_1 \preceq_1 e_2 \wedge e_2 \preceq_2 e_1.$$

Par construction, toute séquence admissible de $E_1 \diamond E_2$ est une séquence admissible de E_1 et de E_2 (intersection de l'ensemble des séquences admissibles).

3.3.3 Observations du système

Le diagnostic d'un système s'appuie sur les observations de ce système.

Définition 3.13 (Observation) *Une* observation *est l'occurrence d'un événement observable de* Σ_{obs}. □

Une observation est associée à un événement observable du modèle. Deux observations peuvent correspondre à un même événement observable, mais ce sont deux occurrences différentes du même événement.

Dans le cadre qui nous concerne, ces observations sont la réception d'alarmes par l'Opérateur de supervision de ce réseau. Dans le chapitre 1, nous avons vu que les alarmes étaient signalisées à l'aide des notifications (voir section 1.4.2.2). Dans ces notifications, il existe en particulier un champ *date* qui informe sur la date d'occurrence de l'alarme. Ainsi, les événements reçus sont datés. Mais de quelle date s'agit-il ? Il en existe de plusieurs types :

– la date d'émission par le composant émetteur ;
– la date de réception par le superviseur.

Cette datation dépend en effet de l'architecture du réseau de gestion et implique des conséquences notables.

Dans le cas où la date est celle d'émission par le composant, étant donnée la nature géographique distribuée du réseau, cette date est établie à partir de l'horloge locale au composant et en aucun cas à partir d'une horloge globale (ce qui nécessiterait des horloges locales synchronisées). Dans ce premier cas, les dates des alarmes ne sont pas une information suffisante pour pouvoir ordonner temporellement l'ensemble des alarmes reçues par le superviseur : il est possible qu'une alarme de date d_1 ait été émise avant une alarme de date d_2 alors que $d_2 < d_1$.

Dans le deuxième cas, la date des alarmes est celle de réception. Ici, on considère que le superviseur est muni d'un capteur qui est connecté au réseau supervisé par des canaux de communications. La date produite par ce capteur impose un ordre total sur les alarmes reçues, par contre il ne s'agit pas d'un ordre sur l'émission des alarmes. Dans ce cas, il faut alors considérer le fait qu'entre l'émission et la réception de l'alarme au superviseur, il y ait un délai qui peut être différent pour chaque type d'alarmes (celles-ci ayant un parcours dans le réseau

qui peut lui être propre), si bien que certaines alarmes peuvent avoir été reçues dans l'ordre inverse de leur émission. Si le modèle du système utilisé prend en compte ce délai, il prend donc en compte ce phénomème d'inversion des alarmes (modèle fondé sur la réception réelle des alarmes). Ceci nécessite de modéliser les canaux de communications entre composants et superviseur (délai, taille, politique). Par contre, pour des raisons d'efficacité, on peut aussi choisir un modèle où l'événement observé est l'émission de l'alarme si bien que le problème d'ordre se pose.

Il existe un troisième cas encore plus complexe : les observations sont datées par un ensemble de capteurs multi-sites associés au superviseur. Dans ce cas, chaque capteur est chargé de recevoir les alarmes d'un sous-ensemble du réseau via des canaux de communications. Chaque capteur est une entité munie de sa propre horloge. Au problème de l'ordre des observations lié à la date de réception s'ajoute alors le problème de la désynchronisation des horloges locales des capteurs qui ne permet pas de dater les réceptions avec un temps global.

En résumé, les alarmes sont reçues séquentiellement par l'Opérateur de supervision munies d'une date qui peut ne pas correspondre à la séquence effectivement reçue pour l'une ou l'autre des raisons citées auparavant. Étant donnés la nature du réseau supervisé, son architecture, on ne peut faire l'hypothèse que cet ordre de réception soit l'ordre dans lequel les composants ont effectivement produit ces alarmes. Par souci de généralité, nous considérons donc dans cette étude théorique, qu'étant donnée une séquence d'observations, nous sommes en mesure d'établir un ordre partiel $\mathcal{O}$ sur ces observations sur lequel sera fondé le diagnostic du système. L'ordre partiel est établi en fonction de l'architecture du réseau (délais de propagation des alarmes, nature de la date associée à une alarme, nature des horloges locales des capteurs multi-sites) et des informations de précédence temporelle, potentiellement disponibles dans les notifications grâce des mécanismes d'estampillage [Fidge 88][Mattern 88].

Définition 3.14 (Comportement observé) *Le comportement observé est un ensemble d'observations $\mathcal{O}$ muni d'une relation d'ordre partiel $\preceq$.* □

En fonction des observations reçues et des informations sur l'observabilité du système, on définit la relation d'ordre $\preceq$ sur les observations. Le comportement observé définit l'ordre d'émissions par les composants élémentaires des observations reçues. Établir le diagnostic du système consiste à établir un diagnostic à partir du modèle et de chaque séquence possible d'observations autorisée par le comportement observé. Dans la suite de ce mémoire, la notation $\mathcal{O}$ sera utilisée pour désigner le comportement observé du système.

Exemple Dans Toynet,

- si on suppose qu'il existe 3 canaux de communications indépendants entre le réseau et un unique capteur associé au centre de supervision (un canal par commutateur),
- si on suppose également que ces canaux ont des délais de propagation bornés par d et que chaque canal se comporte comme une file (*premier arrivé, premier sorti*),

on peut définir une relation d'ordre partiel $\preceq$ s'appuyant sur ces propriétés d'observabilité du système. La séquence d'observations

$$t_0 : SC1op$$
$$t_1 : SC2op$$
$$t_2 : CM3cx31$$
$$t_3 : CM1cx31$$
$$t_4 : CM2blc$$

où les t_i sont les temps de réception donnés par le capteur et $t_0 < t_1 < t_0 + d$, $t_1 + d < t_2$, $t_2 < t_3 < t_2 + d$, $t_3 + d < t_4$ correspond alors à l'ordre partiel présenté sur la figure 3.8.

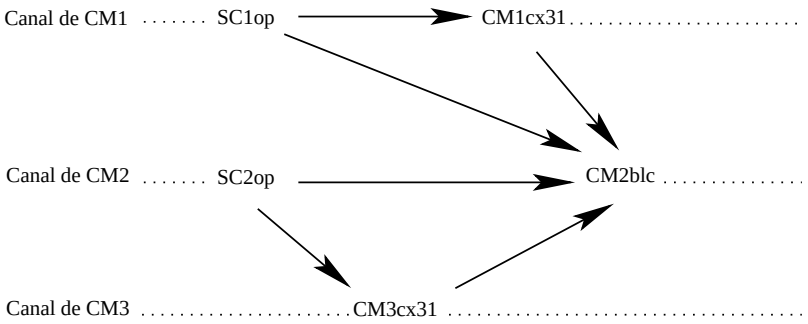

FIG. 3.8 – Comportement observé $\mathcal{O}$.

3.3.4 Comportement observable

Les événements observables sont les événements que l'on peut observer selon le modèle du système à savoir les événements produits $\Sigma_{prod} = \Sigma_{obs}$. À partir du comportement global du système, on peut exprimer son *comportement observable*. Ce comportement correspond à l'ensemble des séquences d'événements observables que le système peut produire selon son modèle.

Définition 3.15 (Comportement observable) *Soit* $\mathcal{C} = q_1 \overset{e_1|\mathcal{I}_1\ \mathcal{O}_1}{\longrightarrow} \dots \overset{e_m|\mathcal{I}_m\ \mathcal{O}_m}{\longrightarrow} q_{m+1}$ *un chemin de transitions de* $\|\Gamma\|$ *(c.-à-d.* $\forall i \in \{1, \dots, m\}, q_i \overset{e_i|\mathcal{I}_i\ \mathcal{O}_i}{\longrightarrow} q_{i+1} \in E$ *où* E *est l'ensemble des transitions de* $\|\Gamma\|$*), le* comportement observable *de* $\mathcal{C}$ *noté* $OBS(\mathcal{C})$ *est l'ensemble partiellement ordonné des occurrences d'événements observables produits par* $\mathcal{C}$ *muni de la relation d'ordre* $\preceq$ *définie par :*

$$\forall i \in \{1, \dots, m\}, \forall o_i \in \mathcal{O}_i, o_i \preceq o_i \quad \textit{(réflexivité) ;}$$

$$\forall i \in \{1, \dots, m-1\}, \forall o_i \in \mathcal{O}_i, \forall o_{i+1} \in \mathcal{O}_{i+1}, o_i \preceq o_{i+1}.$$

Le comportement observable *du modèle du système est l'ensemble des comportements observables de tous les chemins de* $\|\Gamma\|$. □

À un chemin de transitions du modèle, on fait correspondre un ordre partiel sur les occurrences d'événements observables que ce chemin produit. Cet ordre est défini par le fait que

les transitions d'un chemin sont activées séquentiellement : tout événement observable produit par une transition l'est forcément avant tout événement observable produit par les transitions suivantes dans le chemin. Par contre, il est possible qu'une transition synchronisée produise plusieurs événements observables, on considère alors que toute séquence permutation de ces événements peut être observée.

3.3.5 Diagnostic

La définition du diagnostic nécessite l'introduction de certaines notions sur les ensembles partiellement ordonnés.

3.3.5.1 Définition du diagnostic

La définition du diagnostic est fondé sur le modèle de système en question. Cette définition nécessite l'hypothèse de complétude du modèle.

Hypothèse 3.4 (Complétude du modèle) *On suppose que le modèle du système est* complet. □

Le modèle est dit *complet* s'il représente, pour chaque ensemble d'observations pouvant être reçu, l'ensemble des comportements du système qui expliquent ces observations. Autrement dit, pour un comportement observé donné, le modèle contient l'ensemble des informations nécessaires à son diagnostic. D'après cette hypothèse, on a donc la propriété suivante :

Propriété 3.1 *Soit $\mathcal{O}$ le comportement observé du système, il existe au moins un chemin $\mathcal{C}$ de transitions de $\|\Gamma\|$ tel que $\mathcal{O} \diamond OBS(\mathcal{C})$ existe.* □

À partir de cette hypothèse, on peut ainsi définir ce que l'on appelle *diagnostic du système* en fonction du modèle et du comportement observé du système.

Définition 3.16 (Diagnostic) *Soit Γ la description du système, soit $\mathcal{O}$ les observations du système, le diagnostic du système $\Delta(\mathcal{O})$ est l'ensemble des chemins $\mathcal{C}$ de $\|\Gamma\|$ expliquant $\mathcal{O}$, c'est-à-dire tel que $OBS(\mathcal{C}) \diamond \mathcal{O}$ existe.* □

Cette définition exprime le fait que le diagnostic est un ensemble de comportements du système contraints par les observations $\mathcal{O}$. Chaque chemin de transitions est une explication possible du comportement observé. Cette explication est constituée de la séquence d'événements de pannes (primaires et secondaires) ayant eu lieu dans le système ainsi que la propagation de celles-ci. De plus, on y conserve le comportement observable afin d'associer chaque alarme à l'événement dont elle est directement la conséquence. De cette façon, nous conservons l'information complète de l'explication des alarmes. Chaque chemin correspond à un comportement du système. Il est tout à fait possible que le nombre de comportements pouvant expliquer des observations soit infini. Ce cas se produit lorsqu'un nombre non borné d'événéments non-observables peuvent être échangés dans le système entre deux observations.

3.3.6 Difficultés

Un diagnostic est un ensemble de comportements globaux du système. Selon la définition, afin d'établir un tel diagnostic, il faut tout d'abord calculer le comportement global du système. Les systèmes considérés étant de taille importante, le calcul d'un tel comportement global est irréalisable car les ressources informatiques utilisables sont trop limitées.

Taille du comportement global L'exemple de Toynet pose le problème. Ce système est constitué de 12 composants élémentaires. Le nombre maximal d'états dans le modèle d'un composant élémentaire est de 4, et le nombre de transitions est de 13. Le comportement global de ce système très simple est déjà constitué de 5832 états et de 40824 transitions ! Afin d'évaluer cette taille plus formellement, on considère que le système est constitué de n composants élémentaires, chacun étant représenté par un transducteur Γ_i de n_i états ($n_i \geq 2$). Le nombre d'états du comportement global est au pire le nombre d'états du produit libre $\langle \Gamma_1, \ldots, \Gamma_n \rangle$ à savoir $\prod_{i=1}^{n} n_i \geq (min(n_1, \ldots, n_n))^n \geq 2^n$. Autrement dit le nombre d'états au pire du comportement global du système est en $\Omega(2^n)$ où n est le nombre de composants élémentaires du système. Dans les systèmes considérés, ce nombre n est de l'ordre de la centaine, il est donc impossible de calculer et de stocker au moins 2^{100} informations d'états !

3.3.7 Conclusion

Dans le cadre des réseaux de télécommunications, l'information de diagnostic nécessaire à l'Opérateur doit être riche : elle doit permettre de renseigner sur les événements de pannes primaires ainsi que leurs propagations possibles dans le réseau afin d'expliquer le flot d'alarmes reçues. Cette nécessité impose que le diagnostic d'un tel système résume un ensemble de comportements globaux complets du système expliquant le flot d'alarmes, d'où la définition du diagnostic que nous proposons.

Les systèmes que nous considérons sont de taille importante si bien que le comportement global d'un tel système n'est pas implantable. Cette contrainte nous interdit donc l'utilisation de techniques de diagnostic telles que celles présentées dans [Sampath et al. 98], [Sengupta 98], [Debouk et al. 00a]. Bien que les deux dernières soient adaptées aux systèmes distribués, ces approches nécessitent néanmoins la construction du comportement global du système.

Afin d'établir le diagnostic du système, il est donc nécessaire de trouver un moyen grâce auquel il est inutile de passer par la phase de construction du comportement global, nous proposons pour cela la mise en place d'une approche de diagnostic dite *décentralisée*.

3.4 Approche décentralisée

3.4.1 Introduction

L'approche décentralisée du calcul du diagnostic s'appuie sur le principe bien connu de *diviser pour régner*. Ce principe consiste à *briser* un problème en sous-problèmes plus simples de même type, à résoudre chacun de ces sous-problèmes, puis à *fusionner* les résultats obtenus pour apporter une solution au problème posé. Dans le cadre de diagnostic, la division du

problème du diagnostic en sous-problèmes de diagnostic est fondée sur la division du comportement global du système en un ensemble de comportements locaux et la fusion des résultats est un problème de fusion de *diagnostics locaux* afin d'établir le *diagnostic global* du système.

Cette approche est intéressante car elle permet en effet d'établir des résultats intermédiaires fondés sur des modèles de taille raisonnable pour aboutir au même résultat sur le diagnostic du système. Cette section présente donc cette découpe du problème du diagnostic en un ensemble de sous-problèmes par l'introduction de la notion de diagnostic local et par la relation entre les diagnostics locaux et le résultat final.

3.4.2 Décentralisation

Dans une approche fondée sur le principe de *diviser pour régner*, la première chose à effectuer est la découpe du problème en un ensemble de sous-problèmes. Dans le cadre du diagnostic, nous proposons d'effectuer cette découpe par rapport au comportement modélisé. Étant donné le modèle du système $\Gamma = \{\Gamma_1, \ldots, \Gamma_n\}$, on va considérer non plus le comportement global mais un ensemble de comportements locaux.

3.4.2.1 Comportement local

Définition 3.17 (transition localement synchronisée) *Soit un ensemble de modèles de composants élémentaires* $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$*, on dit qu'une transition* $q \xrightarrow{t} q'$ *du produit de* $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k} \rangle$ *est* localement synchronisée *si et seulement si :*

1. $q \xrightarrow{t} q' = (q_{i_1} \xrightarrow{t_{i_1}} q'_{i_1}, \ldots, q_{i_k} \xrightarrow{t_{i_k}} q'_{i_k})$
2. *le cardinal de l'ensemble* $\{t_j | t_j = e_j | \mathcal{E}_j \wedge e_j \in \Sigma_{exo}\}$ *est au plus de 1 ;*
3. *pour chaque* j *de* $\{i_1, \ldots, i_k\}$ *tel que* t_j *est non nulle, on a* $t_j = e_j | \mathcal{E}_j \wedge \forall e \in \mathcal{E}_j \cap (\bigcup_{l=1}^{k} \Sigma_{dec}^{l}), \exists l \in \{i_1, \ldots, i_k\}, t_l = e | \mathcal{E}_l$.

□

Une transition localement synchronisée $t = (t_{i_1}, \ldots, t_{i_k})$ signifie que toutes les transitions t_{i_j} émettant des événements internes à $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$ sont synchronisées avec des transitions t_{i_l} déclenchées par ces événements émis. Une transition localement synchronisée exprime une propagation instantanée d'événements dans $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$.

Propriété 3.2 *Soit* $q \xrightarrow{t} q' = (q_{i_1} \xrightarrow{t_{i_1}} q'_{i_1}, \ldots, q_{i_k} \xrightarrow{t_{i_k}} q'_{i_k})$ *une transition localement synchronisée, on a :*

$$\exists j \in \{i_1, \ldots, i_k\} | (t_j = e_j | \mathcal{E}_j) \wedge e_j \in \Sigma_{dec}^{j} \setminus (\bigcup_{r \in \{i_1, \ldots, i_k\}} \Sigma_{émis}^{r}).$$

□

Démonstration : La transition localement synchronisée $q \xrightarrow{t} q'$ exprime une propagation instantanée d'événements dans $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$. D'après l'hypothèse 3.3, une propagation instantanée d'événements est acyclique, donc il existe au moins un événement dont l'origine n'est pas dans $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$ et qui est la source de cette propagation ; cet événement pouvant être un événement exogène au système ou produit par un composant élémentaire autre que ceux de $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$. □

Une transition localement synchronisée contient ainsi un ensemble de transitions déclenchées par des événements dont l'origine n'appartient pas au groupe de composants $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$. Dans cet ensemble, il existe au plus une transition déclenchée par un événement exogène du système (condition 2). L'ensemble des transitions localement synchronisées du produit libre $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k} \rangle$ définit des contraintes locales de synchronisation.

Notations : les transitions localement synchronisées sont de la forme :

$$(q_{i_1} \xrightarrow{e_{i_1}|\mathcal{I}_{i_1} \cup \mathcal{O}_{i_1}} q'_{i_1}, \ldots, q_{i_k} \xrightarrow{e_{i_k}|\mathcal{I}_{i_k} \cup \mathcal{O}_{i_k}} q'_{i_k}),$$

$\mathcal{I}_{i_j}$ est l'ensemble des événements internes au groupe $\Gamma_{i_1}, \ldots, \Gamma_{i_k}$ dont l'origine est Γ_{i_j} et la cible un autre composant du groupe. $\mathcal{O}_{i_j}$ est l'ensemble des événements observables émis par Γ_{i_j} ainsi que les événements émis par Γ_{i_j} dont la cible est un composant n'appartenant pas à $\Gamma_{i_1}, \ldots, \Gamma_{i_k}$. Tout événement de $\mathcal{I}_{i_j}$ est un événement qui déclenche une transition $q_{i_l} \xrightarrow{e_{i_l}|\mathcal{I}_{i_l} \cup \mathcal{O}_{i_l}} q'_{i_l}, l \neq j$, de plus, dans l'ensemble des événements $\{e_{i_1}, \ldots, e_{i_k}\}$ il existe un sous-ensemble d'événements $\mathcal{E}_{exo}$ exogène au groupe $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$ qui déclenche cette transition. Aussi, afin de simplifier la notation, une telle transition sera exprimée de la façon suivante :

$$(q_{i_1}, \ldots, q_{i_k}) \xrightarrow{\mathcal{E}_{exo}|\mathcal{I}_{i_1} \cup \cdots \cup \mathcal{I}_{i_k}} \mathcal{O}_{i_1} \cup \cdots \cup \mathcal{O}_{i_k} (q'_{i_1}, \ldots, q'_{i_k}).$$

Le *comportement* associé au groupe de composants $\Gamma = \{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$, appelé *comportement local* est défini par un sous-ensemble du produit libre respectant les contraintes de synchronisation locale.

Définition 3.18 (Comportement local) *Le* comportement local *de* $\Gamma_{i_1}, \ldots, \Gamma_{i_k}$ *est le transducteur* (I, O, Q, E') *issu du produit libre* $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k} \rangle = (I, O, Q, E)$ *tel que :*

– $E' \subseteq E$ *est l'ensemble des transitions localement synchronisées de* E.

□

Le comportement local de $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$ représente la réaction instantanée à un stimulus externe du sous-système modélisé par $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$. Ce stimulus correspond à un ensemble d'événements se produisant au même moment (événements pouvant être le résultat d'une propagation de pannes). Ce stimulus contient au plus un événement exogène du système, autrement dit un événement de panne primaire ayant lieu sur des composants modélisés par $\{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$.

Propriété 3.3 *Le comportement local de* $\Gamma = \{\Gamma_1, \ldots, \Gamma_n\}$ *est identique au comportement global* $\|\Gamma\|$. □

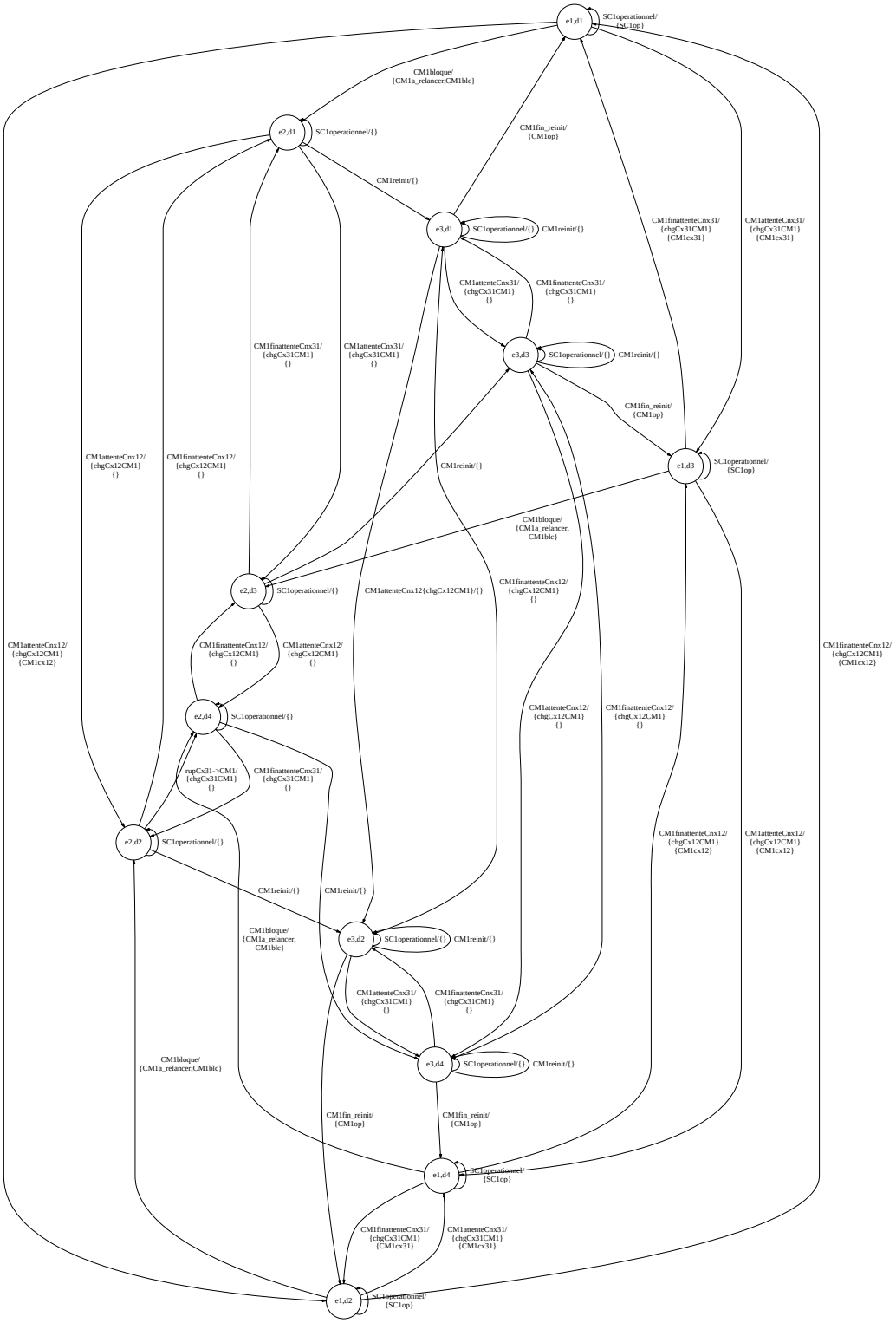

FIG. 3.9 – Comportement local de *CM1ctl* et de *CM1cnx* : $\|CM1ctl, CM1cnx\|$.

Démonstration : Le comportement local de Γ et son comportement global sont issus du même produit libre $\langle \Gamma_1, \ldots, \Gamma_n \rangle$. La seule différence est dans la définition de contraintes de synchronisation des transitions. Il suffit de montrer que dans le cas où l'on considère le groupe de composants $\{\Gamma_1, \ldots, \Gamma_n\}$, toute transition synchronisée est localement synchronisée et réciproquement.

Par définition, toute transition synchronisée de $\|\Gamma\|$ est localement synchronisée (toutes les conditions données dans la définition 3.6 sont contenues dans la définition 3.17). Il nous suffit à présent de montrer qu'une transition localement synchronisée du comportement global de Γ est nécessairement synchronisée.

D'après la définition 3.17, les transitions du comportement local de Γ sont toutes de la forme :

$$(q_1, \ldots, q_n) \xrightarrow{\mathcal{E}_{exo} | \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_n \quad \mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n} (q'_1, \ldots, q'_n)$$

où $\mathcal{E}_{exo}$ est un ensemble d'événements exogènes ou déclenchés par un composant n'appartenant pas à Γ, donc $\mathcal{E}_{exo}$ ne peut contenir que des événements exogènes et donc $card(\mathcal{E}_{exo}) = 1$. En conséquence, d'après la définition 3.6, toute transition localement synchronisée du comportement local de Γ est une transition synchronisée. □

Cette propriété permet donc de parler d'un unique comportement associé à un groupe de composants, qu'il contienne tous les composants élémentaires ou non. Le comportement associé à tout ensemble $\gamma = \{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$ sera noté par la suite $\|\gamma\| = \|\Gamma_{i_1}, \ldots, \Gamma_{i_k}\|$. $\|\gamma\|$ est un transducteur au même titre que le modèle d'un composant élémentaire, au lieu de définir le comportement d'un seul composant élémentaire, il définit le comportement d'un ensemble de ces composants. Par extension, afin d'uniformiser la notation, on dira aussi que $\|\Gamma_i\| = \Gamma_i$ (le comportement de Γ_i est le modèle du composant même).

Définition 3.19 *Soit γ_1 et γ_2 deux ensembles disjoints de composants élémentaires de Γ, on note $\|\|\gamma_1\|, \|\gamma_2\|\|$ le transducteur (I, O, Q, E) issu du produit libre $\langle \|\gamma_1\|, \|\gamma_2\| \rangle$ dont les transitions sont localement synchronisées par rapport à $\gamma_1 \cup \gamma_2$.* □

Avec cette définition, nous étendons l'opération $\|.\|$ à des comportements locaux. La propriété suivante nous permet d'affirmer que son application à des comportements locaux conduit à la construction d'un nouveau comportement local.

Propriété 3.4 *Soit γ_1 et γ_2 deux ensembles disjoints de composants élémentaires de Γ, on a :*

$$\|\gamma_1 \cup \gamma_2\| = \|\|\gamma_1\|, \|\gamma_2\|\|.$$

□

Démonstration : Soit $\gamma_1 = \{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$ et $\gamma_2 = \{\Gamma_{j_1}, \ldots, \Gamma_{j_l}\}$, deux sous-ensembles de composants de Γ distincts. Par définition, $\|\gamma_1 \cup \gamma_2\|$ est un transducteur (I, O, Q, E) issu du produit libre

$$\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k}, \Gamma_{j_1}, \ldots, \Gamma_{j_l} \rangle.$$

De même, $\|\gamma_1\|$ est issu du produit libre $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k} \rangle$ et $\|\gamma_2\|$ est issu du produit libre $\langle \Gamma_{j_1}, \ldots, \Gamma_{j_l} \rangle$. Donc, par définition, $\|\|\gamma_1\|, \|\gamma_2\|\|$ est un transducteur (I', O', Q', E') issu du produit libre

$$\langle \langle \Gamma_{i_1}, \ldots, \Gamma_{i_k} \rangle, \langle \Gamma_{j_1}, \ldots, \Gamma_{j_l} \rangle \rangle.$$

Le produit libre de deux transducteurs étant par définition une opération associative, on en déduit que $\|\|\gamma_1\|, \|\gamma_2\|\|$ est un transducteur issu du même produit libre que $\|\gamma_1 \cup \gamma_2\|$. On a donc $I = I', O = O'$ et

$$Q, Q' \subseteq \prod_{p \in \{i_1, \ldots, i_k, j_1, \ldots, j_l\}} Q_p.$$

Il suffit à présent de montrer que $E = E'$.

E est l'ensemble des transitions localement synchronisées du produit libre $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k}, \Gamma_{j_1}, \ldots, \Gamma_{j_l} \rangle$. Soit E_{γ_1} l'ensemble des transitions de $\|\gamma_1\|$ et E_{γ_2} l'ensemble des transitions de $\|\gamma_2\|$. Chaque transition de E' est une transition localement synchronisée issue du produit des transitions de E_{γ_1} et de E_{γ_2}, ces transitions appartenant au produit libre $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k}, \Gamma_{j_1}, \ldots, \Gamma_{j_l} \rangle$, donc $E' \subseteq E$.

Montrons maintenant que toute transition de E est nécessairement le résultat d'un produit de transitions de E_{γ_1} et de E_{γ_2}. Soit $(q_{i_1} \xrightarrow{t_{i_1}} q'_{i_1}, \ldots, q_{i_k} \xrightarrow{t_{i_k}} q'_{i_k}, q_{j_1} \xrightarrow{t_{j_1}} q'_{j_1}, \ldots, q_{j_l} \xrightarrow{t_{j_l}} q'_{j_l})$ une transition de E où $(q_{i_1} \xrightarrow{t_{i_1}} q'_{i_1}, \ldots, q_{i_k} \xrightarrow{t_{i_k}} q'_{i_k})$ est une transition de $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k} \rangle$ et $(q_{j_1} \xrightarrow{t_{j_1}} q'_{j_1}, \ldots, q_{j_l} \xrightarrow{t_{j_l}} q'_{j_l})$ est une transition de $\langle \Gamma_{j_1}, \ldots, \Gamma_{j_l} \rangle$. Puisque cette transition est localement synchronisée, le cardinal de l'ensemble $\{t_j, (t_j = e_j | \mathcal{E}_j) \wedge e_j \in \Sigma_{exo}\}$ est au plus de 1, donc le cardinal de $\{t_j, (t_j = e_j | \mathcal{E}_j) \wedge j \in \{i_1, \ldots, i_k\} \wedge e_j \in \Sigma_{exo}\}$ est au plus de 1. Pour les mêmes raisons, le cardinal de $\{t_j, (t_j = e_j | \mathcal{E}_j) \wedge j \in \{j_1, \ldots, j_l\} \wedge e_j \in \Sigma_{exo}\}$ est également au plus de 1.

On sait aussi que pour chaque r de $\{i_1, \ldots, i_k, j_1, \ldots, j_l\}$ tel que t_r est non nulle, on a :

$$t_r = e_r | \mathcal{E}_r \wedge \forall e \in \mathcal{E}_r \cap (\bigcup_{v \in \{i_1, \ldots, i_k, j_1, \ldots, j_l\}} \Sigma^v_{dec}), \exists s \in \{i_1, \ldots, i_k, j_1, \ldots, j_l\}, t_s = e | \mathcal{E}_s.$$

En conséquence, si $r \in \{i_1, \ldots, i_k\}$, on a :

$$t_r = e_r | \mathcal{E}_r \wedge \forall e \in \mathcal{E}_r \cap (\bigcup_{v \in \{i_1, \ldots, i_k\}} \Sigma^v_{dec}), \exists s \in \{i_1, \ldots, i_k\}, t_s = e | \mathcal{E}_s$$

et si $r \in \{j_1, \ldots, j_l\}$,

$$t_r = e_r | \mathcal{E}_r \wedge \forall e \in \mathcal{E}_r \cap (\bigcup_{v \in \{j_1, \ldots, j_l\}} \Sigma^v_{dec}), \exists s \in \{j_1, \ldots, j_l\}, t_s = e | \mathcal{E}_s.$$

Par conséquent, $(q_{i_1} \xrightarrow{t_{i_1}} q'_{i_1}, \ldots, q_{i_k} \xrightarrow{t_{i_k}} q'_{i_k})$ est une transition localement synchronisée issue de $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k} \rangle$, elle appartient donc à E_{γ_1}. De même, $(q_{j_1} \xrightarrow{t_{j_1}} q'_{j_1}, \ldots, q_{j_l} \xrightarrow{t_{j_l}} q'_{j_l})$ est une transition localement synchronisée issue de $\langle \Gamma_{j_1}, \ldots, \Gamma_{j_l} \rangle$, elle appartient donc à E_{γ_2}.

Donc $E \subseteq E'$, d'où le résultat. □

Cette propriété exprime le fait que l'on peut obtenir le comportement local d'un ensemble de composants soit à partir des composants élémentaires, soit à partir de comportements locaux issus d'une partition de ces composants élémentaires. Toute opération qui établit le comportement local d'un ensemble de composants à partir d'un ensemble de comportements locaux est donc une opération associative.

Exemple Le comportement local de *SC1*, *CM1ctl*, *CM1cnx* est tel que :

$$\begin{aligned} \|\textit{SC1,CM1ctl,CM1cnx}\| &= \|\textit{SC1,}\|\textit{CM1ctl,CM1cnx}\|\| \\ &= \|\|\textit{SC1,CM1ctl}\|\textit{, CM1cnx}\| \\ &= \|\|\textit{SC1,CM1cnx}\|\textit{, CM1ctl}\|. \end{aligned}$$

3.4.2.2 Choix de la décentralisation

La décentralisation consiste à établir une partition de l'ensemble Γ et à considérer le comportement de chaque élément de cette partition. Le choix d'une partition est une *décentralisation* du modèle.

Exemple Une décentralisation possible de Toynet est la partition suivante : { *SC1*, *CM1ctl*, *CM1cnx* }, { *cnx12* }, { *SC2*, *CM2ctl*, *CM2cnx* }, { *cnx23* }, { *SC3*, *CM3ctl*, *CM3cnx* }, { *cnx31* }. Cette décentralisation est fondée sur la topologie du réseau.

Bien évidemment, le choix de la partition n'est pas unique. Un critère de choix de cette décentralisation est que les comportements locaux sur lesquels on se fonde pour établir les diagnostics locaux soient exploitables car de taille raisonnable. Nous reviendrons dans le chapitre 4 sur d'autres caractéristiques guidant le choix d'une telle décentralisation.

3.4.3 Notion de diagnostic local

Le *diagnostic local* est défini par analogie au diagnostic du système. Au lieu d'être défini à partir du comportement global $\|\Gamma\|$ du système et des observations globales $\mathcal{O}$, il est défini à partir d'un comportement local $\|\gamma\|$ (où $\gamma = \{\Gamma_{i1}, \ldots, \Gamma_{ik}\}$ est un élément de la décentralisation) et des observations locales à $\|\gamma\|$, c'est-à-dire, celles qui ont été émises par les composants $\{\Gamma_{i1}, \ldots, \Gamma_{ik}\}$

Définition 3.20 (Comportement local observé) *Soit $\mathcal{O}$ le comportement observé du système, le* comportement observé local *au sous-système associé à γ est l'ensemble partiellement ordonné $\mathcal{O}_\gamma$ défini par :*

– *$\mathcal{O}_\gamma$ contient toutes les observations de $\mathcal{O}$ émises par des composants élémentaires de γ ;*

– *soit $\preceq_{\mathcal{O}}$ et $\preceq_{\mathcal{O}_\gamma}$ les relations d'ordre respectives de $\mathcal{O}$ et de $\mathcal{O}_\gamma$, on a :*

$$\forall o_1, o_2 \in \mathcal{O}_\gamma, o_1 \preceq_{\mathcal{O}_\gamma} o_2 \equiv o_1 \preceq_{\mathcal{O}} o_2.$$

□

De même, on définit la notion de comportement local observable.

Définition 3.21 (Comportement local observable) *Soit $\mathcal{C} = q_1 \overset{e_1|\mathcal{I}_1\mathcal{O}_1}{\longrightarrow} \dots \overset{e_m|\mathcal{I}_m\mathcal{O}_m}{\longrightarrow} q_{m+1}$ un chemin de transitions de $\|\gamma\|$ (c.-à-d. $\forall i \in \{1,\dots,m\}, q_i \overset{e_i|\mathcal{I}_i\ \mathcal{O}_i}{\longrightarrow} q_{i+1} \in E$ où E est l'ensemble des transitions de $\|\gamma\|$), le* comportement observable *de $\mathcal{C}$ noté $OBS_\gamma(\mathcal{C})$ est l'ensemble partiellement ordonné des occurrences d'événements observables produits par $\mathcal{C}$ muni de la relation d'ordre $\preceq$ définie par :*

$$\forall i \in \{1,\dots,m\}, \forall o_i \in \mathcal{O}_i, o_i \preceq o_i \quad \textit{(réflexivité) ;}$$

$$\forall i \in \{1,\dots,m-1\}, \forall o_i \in \mathcal{O}_i, \forall o_{i+1} \in \mathcal{O}_{i+1}, o_i \preceq o_{i+1}.$$

Le comportement observable *de γ est l'ensemble des comportements observables de tous les chemins de $\|\gamma\|$.*

□

À partir de ces 2 définitions, nous pouvons établir ce que nous entendons par diagnostic local.

Définition 3.22 (Diagnostic local) *Soit $\mathcal{O}_\gamma$ les observations locales à γ, le diagnostic de γ, appelé* diagnostic local *et noté par $\Delta_\gamma(\mathcal{O}_\gamma)$ est l'ensemble des chemins de transitions $\mathcal{C}$ de $\|\gamma\|$ tels que $\mathcal{O}_\gamma \diamond OBS_\gamma(\mathcal{C})$ existe.* □

Le diagnostic local est défini de la même manière que le diagnostic global du système. Il suffit en fait de remplacer les termes $\mathcal{O}_\gamma$ et γ par $\mathcal{O}$ et Γ afin de retrouver la notion de diagnostic du système (le diagnostic du système est le diagnostic local à Γ). Chaque chemin de transitions d'un diagnosic local fournit une explication des observations locales. Cette explication est elle-même locale dans le sens où elle ne contient aucune information sur le comportement des composants voisins de ceux de γ. Par conséquent, une telle explication fait l'hypothèse que toute interaction (échange d'événement) entre un composant de γ et un composant n'appartenant pas à γ est possible.

3.4.4 Diagnostic : fusion des diagnostics locaux

Le problème du calcul du diagnostic global a été subdivisé en un ensemble de calculs de diagnostics locaux fondés sur une décentralisation du modèle. Le problème restant à résoudre est la mise en place de la fusion des résultats afin d'établir le diagnostic du système. Dans la sous-section précédente, il est rappelé que toute explication locale repose sur l'hypothèse que toute interaction entre un composant de γ et un composant n'appartenant pas à γ est possible.

La fusion des résultats en vue de l'établissement du diagnostic global consiste donc à vérifier si les hypothèses d'interactions entre les sous-systèmes proposées par les différents diagnostics locaux sont valides ou non, c'est-à-dire si elles respectent ou non le comportement global du système. Cette vérification d'hypothèses revient à synchroniser les chemins de transitions locaux provenant des différents diagnostics.

Propriété 3.5 *Soit γ_1 et γ_2 deux sous-ensembles disjoints de composants élémentaires du système, tout chemin de transitions de $\Delta_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2})$ est le résultat de la synchronisation d'un chemin de transitions de $\Delta_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et d'un chemin de transitions de $\Delta_{\gamma_2}(\mathcal{O}_{\gamma_2})$.* □

Démonstration : Soit $\mathcal{C}$ un chemin de transitions de $\|\gamma_1 \cup \gamma_2\|$ appartenant au diagnostic $\Delta_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2})$. D'après la propriété 3.4, ce chemin de transitions est issu du produit libre $\langle\|\gamma_1\|, \|\gamma_2\|\rangle$ donc il existe dans $\|\gamma_1\|$ un chemin $\mathcal{C}_1$ et dans $\|\gamma_2\|$ un chemin $\mathcal{C}_2$ tels que $\mathcal{C}$ en est le produit.

Nous allons montrer par l'absurde que $\mathcal{C}_1$ et $\mathcal{C}_2$ appartiennent respectivement à $\Delta_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et à $\Delta_{\gamma_2}(\mathcal{O}_{\gamma_2})$. Supposons que $OBS_{\gamma_1}(\mathcal{C}_1)\diamond\mathcal{O}_{\gamma_1}$ n'existe pas. $\mathcal{C}$ est un chemin de transitions tel que $OBS_{\gamma_1\cup\gamma_2}(\mathcal{C})\diamond\mathcal{O}_{\gamma_1\cup\gamma_2}$ existe ; de plus, il est issu du produit de $\mathcal{C}_1$ et de $\mathcal{C}_2$. Par conséquent, $\mathcal{C}_1$ explique bien toutes les occurrences d'observations de $\mathcal{O}_{\gamma_1}$ mais dans un ordre incompatible avec celui de $\mathcal{O}_{\gamma_1}$ (voir section 3.3.2 page 65). Autrement dit, il existe au moins deux observations différentes o_1 et o_2 telles que $o_1 \preceq o_2$ dans $\mathcal{O}_{\gamma_1}$ et $o_2 \preceq o_1$ dans $OBS_{\gamma_1}(\mathcal{C}_1)$. Puisque o_1 précède o_2 dans $\mathcal{O}_{\gamma_1}$, o_1 précède o_2 dans $\mathcal{O}_{\gamma_1\cup\gamma_2}$ par définition. De même, puisque o_2 précède o_1 dans $OBS_{\gamma_1}(\mathcal{C}_1)$, on a forcément o_2 qui précède o_1 dans $OBS_{\gamma_1\cup\gamma_2}(\mathcal{C})$. En conséquence, $OBS_{\gamma_1\cup\gamma_2}(\mathcal{C}) \diamond \mathcal{O}_{\gamma_1\cup\gamma_2}$ ne peut pas exister, ce qui est contradictoire.

Il en est de même si l'on suppose que $OBS(\mathcal{C}_2) \diamond \mathcal{O}_{\gamma_2}$ n'existe pas. Finalement, puisque que $OBS_{\gamma_1\cup\gamma_2}(\mathcal{C}) \diamond \mathcal{O}_{\gamma_1\cup\gamma_2}$ existe, $OBS(\mathcal{C}_1) \diamond \mathcal{O}_{\gamma_1}$ et $OBS(\mathcal{C}_2) \diamond \mathcal{O}_{\gamma_2}$ sont également définis donc $\mathcal{C}_1$ et $\mathcal{C}_2$ appartiennent respectivement à $\Delta_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et à $\Delta_{\gamma_2}(\mathcal{O}_{\gamma_2})$. □

Cette propriété montre qu'il existe une opération de fusion qui, à partir d'un ensemble de diagnostics locaux, est en mesure d'établir le diagnostic global. La synchronisation des transitions est l'opération nécessaire afin de valider ou non des hypothèses de diagnostic locales.

Corollaire 3.1 *Soit $\{\gamma_1, \ldots, \gamma_m\}$ une décentralisation du système, tout chemin de transitions de $\|\Gamma\|$ appartenant à $\Delta(\mathcal{O})$ est issu de m chemins de transitions, chacun étant issu de $\Delta_{\gamma_i}(\mathcal{O}_{\gamma_i})$* □

Démonstration : On a $\Delta(\mathcal{O}) = \Delta_{\gamma_1\cup\cdots\cup\gamma_m}(\mathcal{O}_{\gamma_1\cup\cdots\cup\gamma_m})$. D'après la propriété 3.5, tout chemin de $\Delta_{\gamma_i\cup\gamma_j}(\mathcal{O}_{\gamma_i\cup\gamma_j})$ est issu de $\Delta_{\gamma_i}(\mathcal{O}_{\gamma_i})$ et de $\Delta_{\gamma_j}(\mathcal{O}_{\gamma_j})$. Si l'on considère un nouveau γ_k, la propriété 3.5 affirme que tout chemin de $\Delta_{\gamma_k\cup\gamma_i\cup\gamma_j}(\mathcal{O}_{\gamma_k\cup\gamma_i\cup\gamma_j})$ est issu d'un chemin de $\Delta_{\gamma_k}(\mathcal{O}_{\gamma_k})$ et d'un chemin de $\Delta_{\gamma_i\cup\gamma_j}(\mathcal{O}_{\gamma_i\cup\gamma_j})$. Par conséquent, tout chemin de $\Delta_{\gamma_k\cup\gamma_i\cup\gamma_j}(\mathcal{O}_{\gamma_k\cup\gamma_i\cup\gamma_j})$ est issu de $\Delta_{\gamma_i}(\mathcal{O}_{\gamma_i})$, $\Delta_{\gamma_j}(\mathcal{O}_{\gamma_j})$ et de $\Delta_{\gamma_k}(\mathcal{O}_{\gamma_k})$. Par extension, on a le résultat. □

3.4.5 Bilan

Le modèle de diagnostic sur lequel s'appuie notre approche est un ensemble de transducteurs $\Gamma = \{\Gamma_1, \ldots, \Gamma_n\}$ communicant à l'aide d'événements. Le diagnostic du système est un ensemble de comportements issus du comportement global $\|\Gamma\|$ (voir figure 3.10). Le problème d'une approche centralisée telle que celles définies dans [Sampath et al. 98] ou encore dans [Rozé 97b] est qu'elle nécessite le calcul de $\|\Gamma\|$ ce qui est impossible dans le cadre des applications qui nous intéressent. L'approche décentralisée, quant à elle, s'appuie sur le principe de *diviser pour régner*. L'idée consiste à construire un ensemble de diagnostics locaux fondés sur des comportements locaux (qui, eux, sont exploitables), puis de fusionner ces diagnostics locaux en vue de construire le diagnostic global. Dans le cadre formel que nous avons mis en place dans ce chapitre, deux opérations sont nécessaires :

1. une *opération de dépliage* qui, étant donnés un comportement $\|\gamma\|$ et un ensemble d'observations $\mathcal{O}_\gamma$ sur ce comportement, établit le comportement compatible avec les observations ;
2. une *opération de fusion* qui établit à partir de deux diagnostics locaux, un diagnostic plus global, cette opération est fondée sur la synchronisation des transitions des modèles locaux.

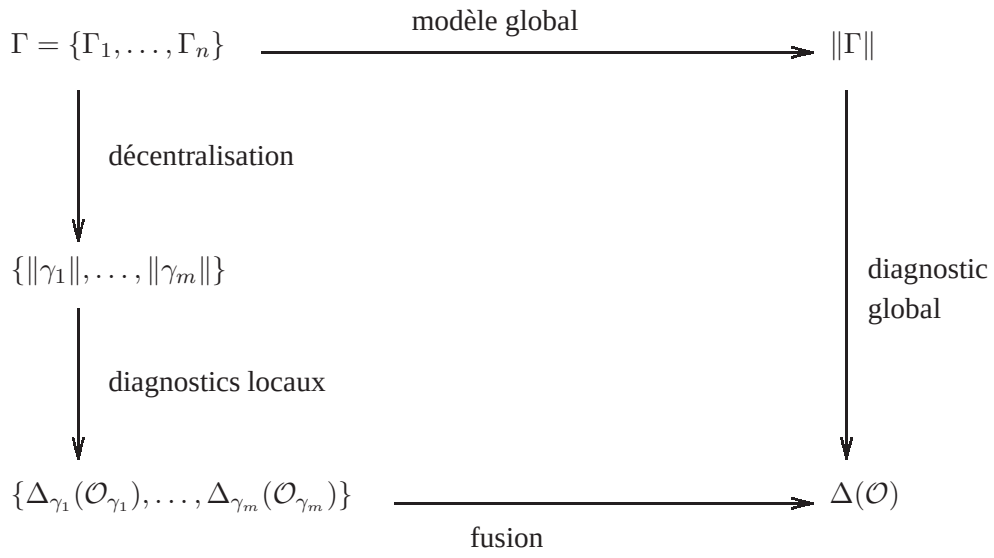

FIG. 3.10 – Approche centralisée / approche décentralisée

La suite de ce mémoire a pour objectif de présenter les choix qui ont été effectués afin de mettre en œuvre l'approche décentralisée, l'objectif principal étant toujours d'avoir des algorithmes les plus efficaces possibles afin d'assurer le calcul en-ligne du diagnostic du système.

Dans le chapitre suivant, nous présentons la mise en œuvre des deux opérations susnommées qui s'appuient sur une représentation efficace des structures mises en jeu dans cette approche. Le chapitre 5, quant à lui, concernera la mise en place d'un algorithme incrémental nécessaire pour assurer l'efficacité de la méthode dans le suivi des observations du système.

CHAPITRE *4*

Mise en œuvre

4.1 Introduction

Dans le chapitre précédent, nous avons établi les bases du diagnostic décentralisé pour des systèmes à événements discrets tels que les réseaux de télécommunications. Ce chapitre a pour objectif de présenter nos choix de mise en œuvre des différents concepts. En particulier, étant donnée la contrainte du suivi en ligne du système supervisé, nos choix ont été guidés par la nécessité d'être efficace.

Ce chapitre présente donc les différents aspects de la mise en œuvre. Dans un premier temps, nous définissons une structure de données représentant un diagnostic. Dans une deuxième section, nous présentons la méthode qui nous permet d'établir à partir d'un comportement local, la structure représentant un diagnostic local. Dans une troisième section, nous présentons la fusion des diagnostics locaux en vue d'obtenir le diagnostic global.

4.2 Représentation des diagnostics

Dans le chapitre précédent, nous avons vu que le diagnostic d'un ensemble de composants élémentaires représentait l'ensemble des comportements de ces composants compatibles avec les observations de ces mêmes composants, chaque comportement étant un chemin de transitions de $\|\Gamma\|$. Le problème est que le nombre de comportements est potentiellement infini : il se peut en effet que le système ait un comportement cyclique lié à des événements non-observables. Une bonne représentation du diagnostic doit disposer d'un moyen de représenter ces comportements plus implicitement. Dans la suite de cette section, nous considérons un ensemble γ de composants élémentaires dont le comportement associé est le transducteur $\|\gamma\| = (I, O, Q, E)$.

4.2.1 Structure de représentation finie

Le diagnostic de γ est un ensemble de chemins de transitions issus de $\|\gamma\|$, aussi peut-on le représenter sous une forme de transducteur comme les modèles.

On note $\|\gamma\|(\mathcal{O}_\gamma) = (I, O, Q', E')$ le transducteur défini par :

– $Q' \subseteq Q \times Pr(\mathcal{O}_\gamma)$ est l'ensemble des états ;

– E' est l'ensemble des transitions entre ces états.

Les états Q' et les transitions E' sont définis de la façon suivante. On considère tous les chemins $\mathcal{C}$ de transitions de $\|\gamma\|$ tels que $\mathcal{O}_\gamma \diamond OBS(\mathcal{C})$ existe. Un tel chemin est de la forme :

$$\mathcal{C} = q_1 \xrightarrow{t_1} \ldots \xrightarrow{t_m} q_{m+1}.$$

Chaque transition $q_i \xrightarrow{t_i} q_{i+1}$ de $\mathcal{C}$ est représentée dans $\|\gamma\|(\mathcal{O}_\gamma)$ de la façon suivante. Si l'on note par $\mathcal{C}_i$ le sous-chemin $\mathcal{C}_i = q_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{i-1}} q_i$, on associe l'état q_i de la transition $q_i \xrightarrow{t_i} q_{i+1}$ à l'état $X_i = (q_i, OBS(\mathcal{C}_i) \diamond \mathcal{O}^i_\gamma) \in Q'$ où $\mathcal{O}^i_\gamma \sqsubseteq \mathcal{O}_\gamma$ tel que $OBS(\mathcal{C}_i) \diamond \mathcal{O}^i_\gamma$ existe. Un tel $\mathcal{O}^i_\gamma$ existe toujours puisque l'on a $\mathcal{O}_\gamma \diamond OBS(\mathcal{C})$ qui existe. De même, on associe l'état q_{i+1} à l'état $X_{i+1} = (q_{i+1}, OBS(\mathcal{C}_{i+1}) \diamond \mathcal{O}^{i+1}_\gamma) \in Q'$. La transition associée dans $\|\gamma\|(\mathcal{O}_\gamma)$ est alors $X_i \xrightarrow{t_i} X_{i+1} \in E'$.

Tout état $(q_1, \emptyset)$ de $\|\gamma\|(\mathcal{O}_\gamma)$ est un *état initial*. Il s'agit d'un état dans lequel l'ensemble des composants de γ peut se trouver avant l'émission de tout événement observable. Tout état $(q_{m+1}, \mathcal{O}_\gamma \diamond OBS(\mathcal{C}))$ de $\|\gamma\|(\mathcal{O}_\gamma)$ est un *état final*. Il s'agit d'un état dans lequel l'ensemble des composants de γ peut se trouver après avoir émis toutes les observations de $\mathcal{O}_\gamma$.

La figure 4.1 présente une partie du diagnostic de $\gamma = \{Cnx12, CM1cnx, CM1ctl, SC1\}$ expliquant les observations $\mathcal{O}_\gamma = \{SC1op, CM1cx12, CM1cx12\}$ avec $SC1op \preceq CM1cx12 \preceq CM1cx12$ représenté par le transducteur $\|\gamma\|(\mathcal{O}_\gamma)$. Cette partie de diagnostic représente l'ensemble des comportements issus de $\|Cnx12, CM1cnx, CM1ctl, SC1\|$ expliquant $SC1op$ suivi de deux occurrences de $CM1cx12$ à partir de l'état (c_1, d_1, e_1, f_1) [1]. Cette figure contient trois états initiaux, ce sont les états dans lesquels les composants de γ peuvent se trouver avant l'émission de la première alarme. L'ensemble des états finals de γ sont représentés par des états à double périphéries, il y en a trois possibles sur la figure. Tout chemin de $\Delta_\gamma(\mathcal{O}_\gamma)$ est représenté par un chemin de transitions de $\|\gamma\|(\mathcal{O}_\gamma)$ dont la source est un état initial et la cible un état final. De même, par construction, tout chemin de transitions de $\|\gamma\|(\mathcal{O}_\gamma)$ entre un état initial et un état final est un chemin de $\Delta_\gamma(\mathcal{O}_\gamma)$. Les transitions émettant les événements observables sont en gras alors que les transitions *silencieuses* sont affichées classiquement. Dans cet exemple, le nombre de comportements expliquant les observations est fini. Dans le cas où ce nombre est infini, des transitions silencieuses forment des cycles entre états de $\|\gamma\|(\mathcal{O}_\gamma)$.

Problème lié à cette représentation Cette représentation, bien qu'étant finie, est sujette à un autre problème : sa taille. En effet, dans ce type de représentation, chaque chemin de transitions représente l'occurrence d'événements en *séquence*. La nature des systèmes étant distribuée, ces systèmes sont en général sujets à des événements concurrents : certaines pannes peuvent se produire indépendamment des autres (par exemple, deux pannes p_1, p_2 qui se produisent sur des composants totalement indépendants l'un de l'autre). Les informations sur les observations dont on dispose ne permettent pas de savoir si telle panne s'est produite avant telle autre. Dans le diagnostic, on voit donc apparaître des hypothèses où p_1 s'est produite avant p_2 et des hypothèses où p_2 s'est produite avant p_1. Il est intéressant de remarquer que dans ce cas, du

[1] Il existe d'autres chemins de transitions expliquant les observations à partir d'autres états, pour des raisons de lisibilté de la figure, nous les avons omis.

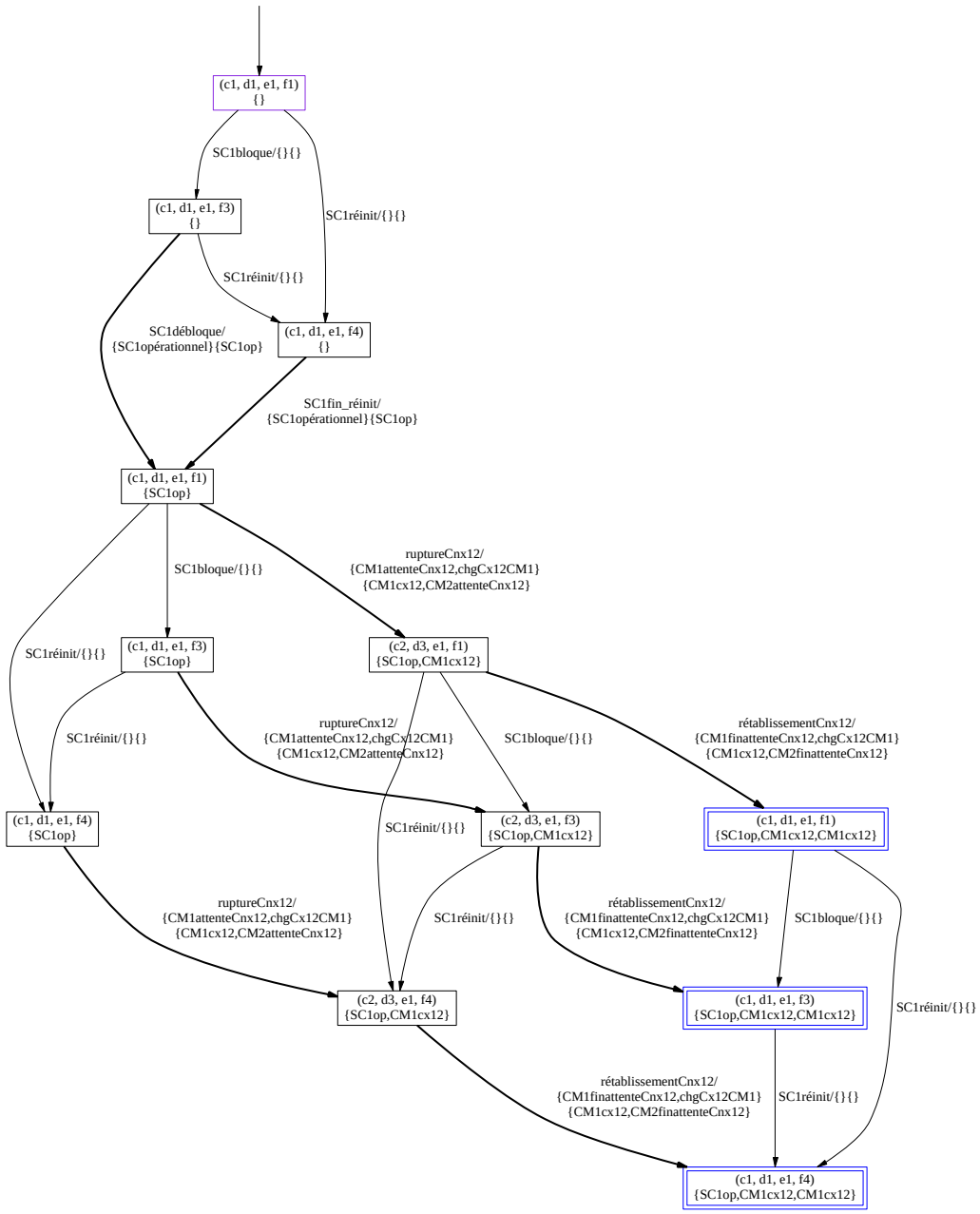

FIG. 4.1 – Partie du transducteur $\|\gamma\|(\mathcal{O}_\gamma)$ représentant le diagnostic local de $\gamma = \{Cnx12, CM1cnx, CM1ctl, SC1\}$ à partir de l'état (c_1, d_1, e_1, f_1) : $\mathcal{O}_\gamma = \{SC1op, CM1cx12, CM1cx12\}$ avec $SC1op \preceq CM1cx12 \preceq CM1cx12$.

point de vue du diagnostic, ce qui est intéressant n'est pas de connaître l'ordre d'occurrence des pannes mais seulement le fait qu'elles aient eu lieu.

Partant de ce constat, un diagnostic peut être vu comme un ensemble de comportements « factorisés », chaque élément de cet ensemble représentant non plus un unique comportement mais un ensemble de comportements « à l'occurrence de pannes indépendantes » près. Ce problème de factorisation des comportements équivalents est bien connu dans le domaine de la vérification de modèle sous le nom du « problème de l'explosion du nombre d'états » et des techniques telles que la réduction d'ordre partiel permettent de résoudre ce problème et donc d'établir des comportements équivalents. Dans la suite de cette section, nous allons présenter la technique de factorisation des comportements en vue de définir une structure de données pour le diagnostic qui s'appuie sur le transducteur $\|\gamma\|(\mathcal{O}_\gamma)$ mais dont la taille est réduite.

4.2.2 Réduction d'ordre partiel

Définition 4.1 (action) *Soit un comportement $\|\gamma\| = (I, O, Q, E)$, l'ensemble des* actions $\mathcal{A}_\gamma$ *est l'ensemble des labels de transitions t tels que $\exists q, q' \in Q, q \xrightarrow{t} q' \in E$.* □

Une action rassemble en particulier la réception d'événements (α) exogènes à γ et l'ensemble (β) des événements produits par γ dont la cible n'est pas un composant de γ. Dans chaque état du comportement, un certain nombre d'actions peuvent se produire, ces actions étant dépendantes de l'état. Par la suite, l'ensemble des états à partir desquels on peut exécuter une action t sera notée $actif_t$.

Définition 4.2 (entrelacement) *Un* entrelacement *d'actions de γ est une séquence finie ou infinie d'actions $v = t_1 t_2 \ldots$ qui génère la séquence d'états $\xi = q_0 q_1 q_2 \ldots$ de Q de longueur $|v| + 1$ (ou ∞ si infinie) telle que :*

- *$\forall i \in \{0, \ldots, |v|\}$, on a $q_i \in actif_{t_{i+1}}$ et $q_i \xrightarrow{t_{i+1}} q_{i+1} \in E$;*
- *v est infinie ou $q_{|v|}$ satisfait $q_{|v|} \notin \bigcup_{t \in \mathcal{A}_\gamma} actif_t$ ($q_{|v|}$ est un état « puits »).*

□

Un entrelacement d'actions est donc une suite maximale d'actions que peut subir la machine γ à partir d'un état initial.

Définition 4.3 (séquence possible) *Une* séquence *d'actions est* possible *si et seulement si elle est un entrelacement ou l'un de ses préfixes.* □

On notera une séquence possible par la séquence de ses actions v. Si la séquence est finie, la séquence fait aboutir γ dans un état final, noté fin_v.

Définition 4.4 (Relation d'indépendance) *Deux actions t_1 et t_2 de $\mathcal{A}_\gamma$ sont indépendantes ssi on a $\forall q \in Q$:*

1. *si $q \in actif_{t_1}$ alors $q \in actif_{t_2}$ si et seulement si $q \xrightarrow{t_1} q' \in E \wedge q' \in actif_{t_2}$;*
2. *si $q \in actif_{t_2}$ alors $q \in actif_{t_1}$ si et seulement si $q \xrightarrow{t_2} q' \in E \wedge q' \in actif_{t_1}$;*

3. si $q \in actif_{t_1} \cap actif_{t_2}$ *alors* $q \xrightarrow{t_2} q' \xrightarrow{t_1} q'' \in E \wedge q \xrightarrow{t_1} q''' \xrightarrow{t_2} q'' \in E$

□

Intuitivement, cette relation d'indépendance entre 2 actions expriment le fait que le déclenchement de l'une d'entre elles n'affecte en rien le déclenchement de l'autre (conditions 1 et 2), de plus, l'ordre dans lequel elles se produisent ne modifie pas l'état atteint au final (condition 3).

À partir de cette relation d'indépendance, on peut exprimer ce qu'est une relation de dépendance :

Définition 4.5 (Relation de dépendance) *Une relation de dépendance* D *est une relation binaire, réflexive et symétrique telle que* $\forall (t_1, t_2) \in D$, t_1 *et* t_2 *ne sont pas indépendantes.* □

Une relation de dépendance D nous permet de définir une relation d'équivalence sur les entrelacements de γ. Dans un premier temps, on définit cette équivalence sur des séquences finies d'actions :

Définition 4.6 *Deux séquences* $v, w \in \mathcal{A}_\gamma^\star$ *sont équivalentes* ($v \equiv_D w$) *s'il existe une séquence* $u_0, u_1, \ldots, u_n$ *où* $u_0 = v$, $u_n = w$ *telle que* $\forall i \in \{0, \ldots, n-1\}, u_i = \overline{u} t_1 t_2 \widehat{u}$ *et* $u_{i+1} = \overline{u} t_2 t_1 \widehat{u}$ *où* $\overline{u}, \widehat{u} \in \mathcal{A}_\gamma^\star$ *et* $(t_1, t_2) \notin D$. □

Ainsi, on dit que v est équivalente à w si l'on peut obtenir w à partir de v par permutations successives d'actions indépendantes.

Exemple soit $v = t_1\, t_2\, t_3\, t_4\, t_5\, t_6$ et $w = t_2\, t_1\, t_3\, t_5\, t_6\, t_4$, on a :

$u_0 = v = t_1\, t_2\, t_3\, t_4\, t_5\, t_6$
$u_1 = t_2\, t_1\, t_3\, t_4\, t_5\, t_6$ (permutation de t_1 et de t_2)
$u_2 = t_2\, t_1\, t_3\, t_5\, t_4\, t_6$ (permutation de t_4 et de t_5)
$u_3 = t_2\, t_1\, t_3\, t_5\, t_6\, t_4 = w$ (permutation de t_4 et de t_6).

Si on a les relations $(t_1, t_2), (t_4, t_5), (t_4, t_6) \notin D$ alors on a $v \equiv_D w$.

Il est facile de voir que la relation $\equiv_D$ est une relation d'équivalence. De plus on peut remarquer que si v est une séquence possible et que $v \equiv_D w$ alors par définition w est aussi une séquence possible.

À présent, nous pouvons étendre la définition d'équivalence entre séquences aux entrelacements. Notons par $Pref(w)$ l'ensemble des préfixes finis de w. La relation $\leq$ entre deux séquences possibles est définie par :

$$v \leq v' \text{ si et seulement si}$$
$$\forall u \in Pref(v) \exists w \in Pref(v') \exists z \in \mathcal{A}_\gamma^\star | w \equiv_D z \wedge u \in Pref(z).$$

Autrement dit, chaque préfixe fini de v est un préfixe d'une permutation (d'opérations indépendantes et adjacentes) d'un préfixe de v'. À l'aide de la relation $\leq$, nous pouvons maintenant définir la relation sur les entrelacements.

Définition 4.7 (équivalence d'ordre partiel) *Soit v, v' deux entrelacements de γ, v est équivalent à v' (noté $v \approx v'$) si et seulement si $v \leq v'$ et $v' \leq v$.* □

Il est facile de voir que $\approx$ est une relation d'équivalence. Cette relation est aussi appelée *relation d'équivalence d'ordre partiel*.

Définition 4.8 (trace) *Une* trace *est une classe d'équivalence définie par la relation $\approx$.* □

Cette définition de trace est aussi appelée *trace de Mazurkiewicz* [Mazurkiewicz 86]. Si nous considérons une séquence admissible v, la trace dont v est membre sera notée $[v]$. Nous pouvons définir la *longueur* d'une trace par la longueur d'une de ses séquences. En effet, par définition, toutes les séquences d'une même trace sont de même taille. Une trace finie contient ainsi l'ensemble des séquences possibles identiques, aux permutations d'opérations adjacentes et indépendantes près. On peut aussi remarquer que toutes les séquences d'une même trace finie ont le même état final. Pour une trace σ, nous noterons l'état final associé fin_σ.

4.2.3 Application de la notion de trace au diagnostic

Le principe de la représentation du diagnostic est le suivant. Une hypothèse de diagnostic est un entrelacement d'actions de γ : une hypothèse de diagnostic fait donc partie d'une trace. Chaque trace représente un ensemble d'hypothèses « factorisées ». Les hypothèses « factorisées » sont des hypothèses de diagnostic pour lesquelles l'ordre d'occurrences de certaines pannes n'a pas d'incidence sur la suite.

D'après la section précédente, il nous suffit de définir une relation de dépendance D qui assurera que les hypothèses de diagnostic sont correctement « factorisées ».

4.2.3.1 Relation de dépendance entre pannes

Nous allons définir une relation de dépendance D_γ sur les actions associées aux transitions de γ qui va nous permettre de définir une représentation réduite du diagnostic. Nous présentons auparavant une notation :

Notation : Soit $t \in \mathcal{A}_\gamma$ une action de γ, soit $\mathcal{E}_t$ l'ensemble des événements liés à t (tous les événements de t émis et reçus, internes ou non), si l'on note γ' un ensemble quelconque de composants élémentaires de Γ alors on notera $\mathcal{I}_{\gamma'}(t) = \{\Gamma_i \in \gamma' | \mathcal{E}_t \cap \Sigma^i_{dec} \neq \emptyset \vee \mathcal{E}_t \cap \Sigma^i_{émis} \neq \emptyset\}$, l'ensemble des composants élémentaires de γ' directement affectés par l'action t.

Définition 4.9 *La relation D_γ est définie par :*

- $D_\gamma \subseteq \mathcal{A}_\gamma \times \mathcal{A}_\gamma$;
- $\forall (t_1, t_2) \notin D_\gamma$, *on a :*
  - $\mathcal{I}_{\Gamma \setminus \gamma}(t_1) = \emptyset \vee \mathcal{I}_{\Gamma \setminus \gamma}(t_2) = \emptyset$;
  - $\mathcal{I}_\gamma(t_1) \cap \mathcal{I}_\gamma(t_2) = \emptyset$;
  - $(\mathcal{E}_{t_1} \cap \Sigma_{obs} = \emptyset \wedge \mathcal{I}_{\Gamma \setminus \gamma}(t_1) = \emptyset) \vee (\mathcal{E}_{t_2} \cap \Sigma_{obs} = \emptyset \wedge \mathcal{I}_{\Gamma \setminus \gamma}(t_2) = \emptyset) \vee (\mathcal{E}_{t_1} \cap \Sigma_{obs} = \mathcal{E}_{t_2} \cap \Sigma_{obs} \wedge \mathcal{I}_{\Gamma \setminus \gamma}(t_1) \cup \mathcal{I}_{\Gamma \setminus \gamma}(t_2) = \emptyset)$.

□

Autrement dit, du point de vue du diagnostic, on considère que deux actions t_1 et t_2 sont dépendantes l'une de l'autre pour une des raisons suivantes :

- t_1 et t_2 peuvent affecter d'autres composants élémentaires que ceux de γ. Dans ce cas, cela signifie que les propagations de pannes peuvent se chevaucher, l'ordre d'activation de t_1 et de t_2 peut donc être discriminant du point de vue du diagnostic.
- t_1 et t_2 affectent des composants élémentaires communs de γ. Il se peut donc que l'activation de t_1 ne soit pas indépendante de l'activation de t_2, il y a là encore une propagation d'événements sur des composants élémentaires communs ;
- t_1 et t_2 émettent des observables toutes les deux ou peuvent conduire à l'émission d'observables (par propagation) et ces observables sont différents. Si les deux actions provoquent l'émission d'observables différents, activer t_1 après t_2 ne produit pas le même comportement observable que d'activer t_2 après t_1, ces deux actions ne sont donc pas indépendantes du point de vue du diagnostic.

Propriété 4.1 *La relation D_γ est une relation de dépendance.* □

Démonstration : Il faut montrer que la relation D_γ est une relation binaire réflexive et symétrique et que toutes actions t_1, t_2 telles que $(t_1, t_2) \notin D_\gamma$ sont indépendantes selon la définition 4.4.

1. **Réflexivité :** pour tout $t \in \mathcal{A}_\gamma$, $\mathcal{I}_\gamma(t)$ contient au moins un composant élémentaire associé à γ, donc $\mathcal{I}_\gamma(t) \neq \emptyset$. La relation est réflexive.
2. **Symétrie :** par définition, la relation est symétrique.
3. **Critère d'indépendance :** il nous suffit de montrer que si $(t_1, t_2) \notin D_\gamma$ alors t_1 et t_2 sont indépendants au sens de la définition 4.4. Supposons que $(t_1, t_2) \notin D_\gamma$. Supposons également que $\|\gamma\|$ soit le comportement local de k composants élémentaires $\Gamma_{i_1}, \ldots, \Gamma_{i_k}$ du système : soit $q = (q_{i_1}, \ldots, q_{i_k})$ un état de $\|\gamma\|$.

   1) Montrer que si $q \in actif_{t_1}$, on a $q \in actif_{t_2} \Rightarrow (q \xrightarrow{t_1} q' \in E \wedge q' \in actif_{t_2})$

   Supposons maintenant que $q \in actif_{t_1}$, autrement dit dans $\|\gamma\|$, on a une transition localement synchronisée

   $$(q_{i_1} \xrightarrow{t_1^{i_1}} q'_{i_1}, \ldots, q_{i_k} \xrightarrow{t_1^{i_k}} q'_{i_k}).$$

   De même, supposons que $q \in actif_{t_2}$, autrement dit dans $\|\gamma\|$, on a une transition localement synchronisée

   $$(q_{i_1} \xrightarrow{t_2^{i_1}} q''_{i_1}, \ldots, q_{i_k} \xrightarrow{t_2^{i_k}} q''_{i_k}).$$

   Puisque $(t_1, t_2) \notin D_\gamma$, on a $\mathcal{I}_\gamma(t_1) \cap \mathcal{I}_\gamma(t_2) = \emptyset$, ce qui implique qu'il existe un sous-ensemble $(q_{i_j})_{j \in J \subseteq \{1,\ldots,k\}}$ d'états de q qui n'ont pas été changés par l'action t_1 (présence de transitions nulles $q_{i_j} \xrightarrow{\mathbf{e}|\{\}} q_{i_j}$) et que seuls des états de $(q_{i_j})_{j \in J \subseteq \{1,\ldots,k\}}$ sont sources d'une transition non nulle dans l'activation de t_2. Soit $q' = (q'_{i_1}, \ldots, q'_{i_k})$ et soit la transition :

$$q' \xrightarrow{t_2} q''' = (q'_{i_1} \xrightarrow{t_2^{i_1}} q'''_{i_1}, \ldots, q'_{i_k} \xrightarrow{t_2^{i_k}} q'''_{i_k}).$$

Comme $(q_{i_j})_{j\in J\subseteq\{1,\ldots,k\}}$ sont nécessairement sources d'une transition nulle dans l'activation de t_1 donc on a : $(q_{i_j})_{j\in J\subseteq\{1,\ldots,k\}} = (q'_{i_j})_{j\in J\subseteq\{1,\ldots,k\}}$. Comme seuls des états de $(q_{i_j})_{j\in J\subseteq\{1,\ldots,k\}}$ sont sources d'une transition non nulle dans l'activation de t_2, on a nécessairement :

– toute transition $q'_{i_l} \xrightarrow{t_2^{i_l}} q'''_{i_l}$ où q'_{i_l} ne fait pas partie des (q'_{i_j}) est une transition nulle ;

– toute transition $q'_{i_l} \xrightarrow{t_2^{i_l}} q'''_{i_l}$ où q'_{i_l} fait partie des (q'_{i_j}) est soit nulle, soit de la forme $q_{i_j} \xrightarrow{t_2^{i_j}} q'''_{i_j}$.

Comme $\|\gamma\|$ est issu du produit libre $\langle \Gamma_{i_1}, \ldots, \Gamma_{i_k}\rangle$, la transition $q' \xrightarrow{t_2} q'''$ est forcément une transition de ce produit. De plus, comme $q \in actif_{t_2}$, $q' \xrightarrow{t_2} q'''$ est nécessairement localement synchronisée, elle appartient donc à $\|\gamma\|$, donc $q' \in actif_{t_2}$.

2) Montrer que si $q \in actif_{t_1}$**, on a** $(q \xrightarrow{t_1} q' \in E \wedge q' \in actif_{t_2}) \Rightarrow q \in actif_{t_2}$

Soit q et q' tels que $(q \xrightarrow{t_1} q' \in E \wedge q' \in actif_{t_2})$, $q \xrightarrow{t_1} q'$ est de la forme

$$(q_{i_1} \xrightarrow{t_1^{i_1}} q'_{i_1}, \ldots, q_{i_k} \xrightarrow{t_1^{i_k}} q'_{i_k}).$$

Puisque $q' \in actif_{t_2}$, il existe un ensemble $(q_{i_j})_{j\in J\subseteq\{1,\ldots,k\}} = (q'_{i_j})_{j\in J\subseteq\{1,\ldots,k\}}$ car $\mathcal{I}_\gamma(t_1) \cap \mathcal{I}_\gamma(t_2) = \emptyset$. Soit la transition,

$$q \xrightarrow{t_2} q'' = (q_{i_1} \xrightarrow{t_2^{i_1}} q''_{i_1}, \ldots, q_{i_k} \xrightarrow{t_2^{i_k}} q''_{i_k}),$$

on montre de la même manière que précédemment qu'une telle transition est dans $\|\gamma\|$ et donc que $q \in actif_{t_2}$.

Le critère d'indépendance 1 est donc vérifié. Le critère d'indépendance 2 s'obtient par une démonstration identique en échangeant t_1 avec t_2.

3) Montrer que si $q \in actif_{t_1} \cap actif_{t_2}$ **alors** $q \xrightarrow{t_2} q' \xrightarrow{t_1} q'' \in E \wedge q \xrightarrow{t_1} q''' \xrightarrow{t_2} q'' \in E$

On montre facilement le résultat en présentant le fait que l'activation de t_1 puis de t_2 aboutit nécessairement dans un état q'' tel que il existe deux sous-ensembles I_1 et I_2 distincts de $\{1,\ldots,k\}$ tels que $q''_{i_j} = q'''_{i_j}, \forall j \in I_1$, $q''_{i_j} = q'_{i_j}, \forall j \in I_2$ et $q''_{i_j} = q_{i_j}, \forall j \in \{1,\ldots,k\} \setminus (I_1 \cup I_2)$. Si on active t_2 avant t_1, on aboutit à ce même état.

□

4.2.4 Représentation réduite du diagnostic

À l'aide de la relation de dépendance D_γ, il est possible de construire un transducteur réduit qui exprime le même comportement que $\|\gamma\|(\mathcal{O}_\gamma)$. Nous appelerons $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$, le *diagnostic réduit* de γ expliquant les observations $\mathcal{O}_\gamma$. Ce diagnostic réduit est défini de la façon suivante.

Définition 4.10 (Diagnostic réduit) *La* réduction *du diagnostic* $\Delta_\gamma(\mathcal{O}_\gamma) = \|\gamma\|(\mathcal{O}_\gamma) = (I, O, Q, E)$ *est un transducteur* $\Delta_\gamma^{red}(\mathcal{O}_\gamma) \triangleq (I, O, Q', E')$ *où*

- $Q' \subseteq Q$ *est l'ensemble des états ;*
- $E' \subseteq E$ *est l'ensemble des transitions. Chaque chemin de transitions* $q_0 \xrightarrow{t_1} q_1 \dots q_{m-1} \xrightarrow{t_m} q_m$ *est l'unique représentant de la trace* $[t_1 \dots t_m]$ *définie par la relation de dépendance* D_γ.

□

La figure 4.2 présente la réduction du diagnostic des composants $\gamma = \{Cnx12, CM1cnx, CM1ctl, SC1\}$ expliquant les observations $\mathcal{O}_\gamma = \{SC1op, CM1cx12, CM1cx12\}$ avec $SC1op \preceq CM1cx12 \preceq CM1cx12$ dont la représentation non-réduite est sur la figure 4.1. Cette réduction profite du fait que le blocage de la station représenté par les étiquettes de transitions du type $t_1 = SC1bloque/\{\}\{\}$ ou sa réinitialisation $t_2 = SC1reinit/\{\}\{\}$ sont indépendantes de la rupture de connexion de $Cnx12$ représentée par les étiquettes de transitions du type $t_3 = ruptureCnx12/\{\dots\}\{\dots\}$ et de son rétablissement $t_4 = retablissementCnx12/\{\dots\}\{\dots\}$. On a $(t_1, t_3), (t_2, t_3), (t_1, t_4), (t_2, t_4) \notin D_\gamma$. Chaque chemin de transitions représente une trace d'événements, il représente un ensemble de chemins de transitions de la figure 4.1.

Canonicité de la représentation

D'après cette définition, il existe plusieurs diagnostics réduits pouvant représenter un même diagnostic. Cela est dû au fait que tout chemin de transitions représenté par une même trace est un bon candidat pour représenter cette trace. Afin d'avoir une représentation canonique d'un diagnostic, il suffit d'imposer un ordre total sur les actions (l'ordre lexicographique par exemple) et de considérer comme représentant d'une trace, le chemin de transitions qui viole le moins cet ordre sur les actions.

Exemple Soit la trace $[dabcghe]$ telle que $(a, d), (b, d), (e, h) \notin D_\gamma$, l'ensemble des séquences possibles est : $dabcghe$, $adbcghe$, $abdcghe$, $adbcgeh$, $dabcgeh$, $abdcgeh$. Afin d'assurer la canonicité de la représentation du diagnostic, on priviligiera comme représentant de la trace $[dabcghe]$ le chemin de transitions qui est associé à la séquence $abdcgeh$.

Dans la suite de ce chapitre, la construction d'un tel diagnostic est présentée :

- dans un premier temps, l'opération de dépliage permettant la construction d'un diagnostic local réduit ;
- puis, l'opération de fusion des diagnostics réduits en vue d'obtenir un nouveau diagnostic réduit plus global.

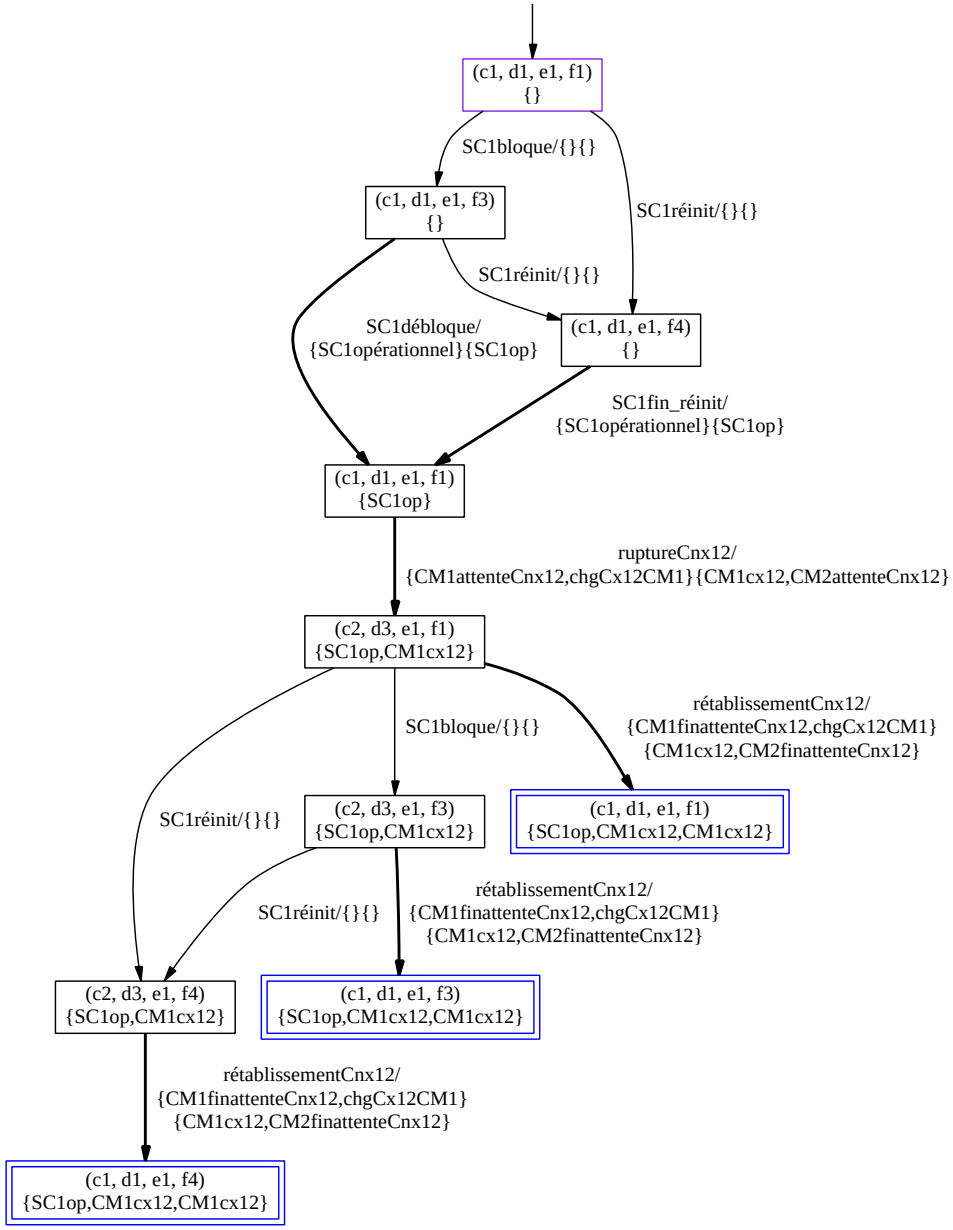

FIG. 4.2 – Représentation réduite du diagnostic présenté sur la figure 4.1.

4.3 Diagnostic local

Pour construire un diagnostic local $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$, l'algorithme doit identifier dans le transducteur γ les chemins de transitions qui expliquent les observations. Dans cette section, plusieurs algorithmes pour le calcul d'un diagnostic local réduit sont présentés.

Hypothèse 4.1 *Le diagnostic local est établi sur un flux unique d'observations : l'ensemble des observations $\mathcal{O}_\gamma$ constitue un ordre total.* □

Les algorithmes présentés ci-dessous se rapporteront à cette hypothèse. En fin de cette section, nous décrirons une extension des algorithmes qui ne nécessite pas cette hypothèse, avec les difficultés supplémentaires qu'elle doit résoudre. Cette hypothèse exprime le fait que nous cherchons à déterminer les chemins de transitions $\mathcal{C}$ de γ tels que $OBS_\gamma(\mathcal{C}) \diamond \mathcal{O}_\gamma$ existe. Puisque $\mathcal{O}_\gamma$ constitue un ordre total cela revient à dire que $OBS_\gamma(\mathcal{C}) \diamond \mathcal{O}_\gamma = \mathcal{O}_\gamma$.

4.3.1 Principe

Le principe du calcul du diagnostic $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ est le suivant. À partir de γ et d'un ensemble d'observations $\mathcal{O}_\gamma$, il faut établir tous les chemins de transitions qui représentent toutes les traces possibles contenues dans le transducteur $\|\gamma\|(\mathcal{O}_\gamma)$. Le calcul du diagnostic local est un processus incrémental : on considère un diagnostic $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ et une nouvelle observation o qui constitue avec $\mathcal{O}_\gamma$ un nouvel ensemble d'observations $\mathcal{O}'_\gamma$, l'objectif est alors d'adapter $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ en vue d'obtenir $\Delta_\gamma^{red}(\mathcal{O}'_\gamma)$. Le calcul du diagnostic local peut s'effectuer à l'aide d'un algorithme de recherche en profondeur d'abord. L'algorithme que nous proposons est adapté de [Godefroid et Wolper 91] et [Peled 93], l'idée est de parcourir intelligemment l'espace d'états définis par $\|\gamma\|(\mathcal{O}'_\gamma)$ afin de calculer toutes les traces possibles et de ne retenir qu'un chemin de transitions unique comme représentant d'une trace.

4.3.2 Exploration réduite d'un état

La brique de base pour le calcul d'un diagnostic local est l'exploration réduite des chemins de transitions issus d'un état de $\|\gamma\|$ qui expliquent une observation o. Les algorithmes 1 et 2 présentent cette exploration : il s'agit d'une adaptation des algorithmes décrits dans [Godefroid et Wolper 91] et [Peled 93]. Le résultat de ces algorithmes est la construction de l'ensemble des transitions $DiagRed(q_0, e, o)$ où q_0 est un état de $\|\gamma\|$, e est un ensemble d'actions de $\mathcal{A}_\gamma$ et o est l'observation à expliquer à partir de l'état q_0.

$DiagRed(q_0, e, o)$ contient l'ensemble des transitions de $\|\gamma\|$ qui constituent les chemins $\mathcal{C} = q_0 \xrightarrow{t_0} q_1 \dots q_m \xrightarrow{t_m} q_{m+1}$ tels que :

1. $OBS_\gamma(q_0 \xrightarrow{t_0} q_1 \dots q_{m-1} \xrightarrow{t_{m-1}} q_m) = \emptyset$;
2. $o \in OBS_\gamma(q_m \xrightarrow{t_m} q_{m+1})$;
3. $\mathcal{C}$ est l'unique représentant de $[t_0, \dots, t_m]$ issu de q_0 ;
4. $\forall t \in e, t = t_i \Rightarrow (\exists j \in \{0, \dots, i-1\} | (t_j, t_i) \in D)$.

Algorithme 1 Calcul de $DiagRed(q_0, e, o)$.

Entrée 1 : $\|\gamma\| = (I, O, Q, E)$

Entrée 2 : $q_0 \in Q$

Entrée 3 : $e \subseteq \mathcal{A}_\gamma$

Entrée 4 : $o \in \Sigma_{obs}^\gamma$

explorées$(q_0) \leftarrow \emptyset$ *; incertaines*$(q_0) \leftarrow \emptyset$ *; à_visiter*$(q_0) \leftarrow \mathbf{vrai}$ *; évitées*$(q_0) \leftarrow e$

pour tout $q \in Q$ **faire**

statut$(q) \leftarrow$ *en_test*; *évitées2*$(q) \leftarrow \emptyset$

fin pour

atteintes $\leftarrow$ *visiter_état*(q_0)

si $\nexists q \xrightarrow{t} q' \in$ *atteintes*$|o \in OBS_\gamma(q \xrightarrow{t} q')$ **alors**

atteintes $\leftarrow \emptyset$ {Il n' y a pas de solution.}

sinon

pour tout $q \xrightarrow{t} q' \in$ *atteintes*$|$*statut*$(q) =$ *possible* $\wedge$ *statut*$(q') =$ *fixé* **faire**

statut$(q) \leftarrow$ *fixé* {Un état possible, source d'une transition dont la cible est fixée, est un état fixé}

fin pour

pour tout $q \xrightarrow{t} q' \in$ *atteintes*$|\neg($*statut*$(q) =$ *fixé* $\wedge$ *statut*$(q') =$ *fixé*$)$ **faire**

atteintes $\leftarrow$ *atteintes* $\setminus \{q \xrightarrow{t} q'\}$ {Élimination des transitions dont un des états n'est pas fixé}

fin pour

fin si

Sortie : $DiagRed(q_0, e, o) \leftarrow$ *atteintes*

Les conditions 1 et 2 signifient que $DiagRed(q_0, e, o)$ contient des chemins de transitions issus de q_0 expliquant un comportement non-observable suivi d'une transition émettant l'observable o. La condition 3 assure que les chemins retenus sont les uniques représentants de leurs traces respectives. La condition 4 est fondée sur l'ensemble des actions de l'ensemble e et assure que si une trace calculée contient une action t appartenant à e, alors cette action apparaît dans la trace uniquement après une action qui est dépendante de t (l'usage d'un tel ensemble d'actions sera expliqué dans le paragraphe suivant).

Commentaires sur l'algorithme de construction de $DiagRed(q_0, e, o)$

L'algorithme est un parcours en profondeur du transducteur associé à $\|\gamma\|$. Il s'agit d'un parcours « intelligent » détectant une transition observable expliquant o. Lors du retour-arrière (*backtrack*), on retient les transitions pouvant appartenir à $DiagRed(q_0, e, o)$.

Pour chaque état q exploré à partir de q_0, on gère les ensembles suivants :
- *explorées(q)* contient l'ensemble des actions issues de q qui ont été explorées à un instant donné ;
- *évitées(q)* et *évitées2(q)* contiennent des actions qu'il est inutile de développer à partir de q ;
- *incertaines(q)* contient l'ensemble des actions pour lesquelles on ne peut pas statuer (faut-il les éviter ou non ?) ;
- *statut(q)* contient le statut de l'état q : *fixé* signifie que q fait partie d'un chemin de $DiagRed(q_0, e, o)$, *possible* signifie que q fait partie d'un chemin de $DiagRed(q_0, e, o)$ sous réserve que les successeurs de q dans ce chemin soient fixés, *en_test* signifie en fin d'algorithme que q n'appartient pas à un chemin de $DiagRed(q_0, e, o)$;
- *état_visité(q)* rend vrai si l'état est en cours d'exploration, faux sinon ;
- *à_visiter(q)* rend vrai si l'état est à explorer, faux sinon.

À côté de ces ensembles, l'algorithme gère les structures suivantes.
- *dépendantes(t)* est l'ensemble des actions dépendantes de t, à savoir :

$$\textit{dépendantes(t)} \triangleq \{t' \in \mathcal{A}_\gamma | (t, t') \in D\}$$

- *atteintes* est l'ensemble des transitions candidates à un instant donné pour appartenir à $DiagRed(q_0, e, o)$.

Le principe de l'algorithme est le suivant. Après quelques initialisations de structures, on débute le parcours de $\|\gamma\|$ à partir de l'état q_0 (*visiter_état*(q_0), algorithme 1). Pour chaque état visité (voir algorithme 2, lignes 3-4), on regarde les actions t qui en sont issues et qui ne peuvent pas être évitées, certaines étant non-observables (*actions_non_obs_à_traiter*) et d'autres émettant l'observation o (*actions_obs_à_traiter*) [2]. On traite chaque action dans un ordre par-

[2] Il existe aussi des actions observables issues de q mais n'émettant pas l'observation o, celles-ci ne sont pas considérées, les transitions correspondantes ne peuvent pas faire partie de $DiagRed(q_0, e, o)$.

ticulier, les actions non-observables puis les actions observables. Dans chaque ensemble les actions sont extraites dans un ordre bien défini (l'ordre lexicographique par exemple) (lignes 6 et 34).

Pour chaque action t non observable traitée, on établit l'ensemble des actions qu'il est inutile de parcourir à partir de l'état cible q' de cette transition (ligne 8). Cet ensemble d'actions est constitué des actions à éviter dans l'état courant et des actions déjà parcourues à partir de l'état courant qui sont indépendantes de l'action t associée à la transition courante :

$$\textit{à\_éviter} \leftarrow (\textit{evitées}(q) \cup \textit{explorées}(q)) \setminus (\textit{incertaines}(q) \cup \textit{dépendantes}(t)).$$

Si l'état q' n'a pas été exploré, on débute son exploration (lignes 10-11). Si l'état q' a déjà été exploré, cet état est déjà associé à un ensemble d'actions à éviter. Si cet ensemble ne contient pas le nouvel ensemble *à_éviter*, il faut réexplorer cet état (lignes 12-20).

Si l'état q' a un statut fixé ou possible, la transition courante fait donc partie des transitions atteintes et q prend le statut de q'. Si q' est en cours d'exploration, cela signifie que l'on a détecté un cycle de transitions, l'action t issue de q est jugée incertaine car on ne sait pas si la transition courante sera en définitive une transition atteinte : on l'ajoute, on devra l'enlever si à la fin de l'algorithme, l'état q' n'est pas fixé (lignes 21-31).

Dans le cas où l'action t est observable, le traitement est différent. En effet, si l'action est observable, cela signifie que l'on a trouvé un chemin de transitions à partir de q_0 expliquant l'observation o. L'état q' est donc fixé de même que l'état q. Il est inutile d'explorer q'. Par contre, on conserve l'ensemble des actions à éviter concernant l'exploration de q' (lignes 33-39) dans un nouvel ensemble *évitées2(q)*.

Une fois l'exploration de q_0 effectuée (voir algorithme 1), l'ensemble des transitions de $DiagRed(q_0, e, o)$ est inclus dans l'ensemble *atteintes*. Si parmi ces transitions aucune n'émet l'observation o, cela signifie qu'il n'y a pas de solution, $DiagRed(q_0, e, o) = \emptyset$. Dans le cas contraire, on fixe les états qui peuvent l'être (ils sont les prédécesseurs d'états fixés dans une transition atteinte) et on supprime de l'ensemble *atteintes* les transitions dont les états cible et source ne sont pas fixés. Après cette élimination, $DiagRed(q_0, e, o) = \textit{atteintes}$.

La figure 4.3 présente les chemins de transitions de $DiagRed((c_2, d_3, e_1, f_1), \emptyset, CM1cx12)$. (c_2, d_3, e_1, f_1) est un état de $\|Cnx12, CM1cnx, CM1ctl, SC1\|$ (voir les figures 4.1 et 4.2). Chaque chemin de transitions entre (c_2, d_3, e_1, f_1) et un état cible d'une transition observable représente une trace expliquant $CM1cx12$ à partir de l'état (c_2, d_3, e_1, f_1) de $\|Cnx12, CM1cnx, CM1ctl, SC1\|$.

Par la suite, on considérera aussi l'ensemble $DiagRed(q_0, e)$ qui recense l'ensemble des traces non-observables issues de q_0. Son calcul est presque identique à celui de $DiagRed(q_0, e, o)$. En effet, dans ce calcul, on ne considère que les actions non-observables (*actions_obs_a_traiter* $= \emptyset$). Un état est fixé s'il est prédécesseur d'un état fixé (comme dans l'algorithme précédent) ou s'il existe une action observable issue de cet état (action qui n'est pas activée).

Algorithme 2 Exploration réduite d'un état.

Fonction *visiter_état*(q)
état_visité$(q) \leftarrow \mathbf{vrai}$
actions_non_obs_à_traiter$(q) \leftarrow \{t | q \xrightarrow{t} q' \in E \wedge OBS_\gamma(q \xrightarrow{t} q') = \emptyset\} \setminus$ *évitées*(q)
actions_obs_à_traiter$(q) \leftarrow \{t | q \xrightarrow{t} q' \in E \wedge o \in OBS_\gamma(q \xrightarrow{t} q')\} \setminus$ *évitées*(q)
tant que *actions_non_obs_à_traiter*$(q) \neq \emptyset$ **faire**
  $t \leftarrow$ *oter_action*(*actions_non_obs_à_traiter*(q)) ; *explorées*$(q) \leftarrow$ *explorées*$(q) \cup \{t\}$
  Soit $q' | q \xrightarrow{t} q' \in E$; *à_éviter* $\leftarrow$ (*évitées*$(q) \cup$ *explorées*$(q)) \setminus$ (*incertaines*$(q) \cup$ *dépendantes*(t))
  si $\neg$*état_visité*(q') **alors**
    explorées$(q') \leftarrow \emptyset$; *incertaines*$(q') \leftarrow \emptyset$; *évitées*$(q') \leftarrow$ *à_éviter*
    à_visiter$(q') \leftarrow \mathbf{vrai}$; *atteintes* $\leftarrow$ *atteintes* $\cup$ *visiter_état*(q')
  sinon si *évitées*$(q') \neq \emptyset \wedge$ *à_éviter* $\not\subseteq$ *évitées*(q') **alors**
    explorées$(q') \leftarrow \emptyset$; *incertaines*$(q') \leftarrow \emptyset$;
    si $\neg$*à_visiter*(q') **alors**
      évitées$(q') \leftarrow$ évitées$(q') \cap$ à_éviter
      à_visiter$(q') \leftarrow \mathbf{vrai}$; *atteintes* $\leftarrow$ *atteintes* $\cup$ *visiter_état*(q')
    sinon
      actions_non_obs_à_traiter$(q') \leftarrow$ *actions_non_obs_à_traiter*$(q') \cup ((\text{évitées}(q') \setminus \text{à\_éviter}) \cap \{t | q' \xrightarrow{t} q'' \in E \wedge OBS_\gamma(q' \xrightarrow{t} q'') = \emptyset\})$
      actions_obs_à_traiter$(q') \leftarrow$ *actions_obs_à_traiter*$(q') \cup ((\text{évitées}(q') \setminus \text{à\_éviter}) \cap \{t | q' \xrightarrow{t} q'' \in E \wedge o \in OBS_\gamma(q' \xrightarrow{t} q'')\})$
      évitées$(q') \leftarrow$ *évitées*$(q') \cap$ *à_éviter*
    fin si
  fin si
  si *statut*$(q') \in \{$*fixé*,*possible*$\} \vee$ *à_visiter*(q') **alors**
    atteintes $\leftarrow$ *atteintes* $\cup \{q \xrightarrow{t} q'\}$
    si *à_visiter*(q') **alors**
      incertaines$(q) \leftarrow$ *incertaines*$(q) \cup \{t\}$
    fin si
    si *statut*$(q') \in \{$*fixé*,*possible*$\}$ **alors**
      statut$(q) \leftarrow$ *statut*(q')
    sinon
      statut$(q) \leftarrow$ *possible* $\{q \xrightarrow{t} q'$ termine un cycle et q' est en cours de test.$\}$
    fin si
  fin si
fin tant que
tant que *actions_obs_à_traiter*$(q) \neq \emptyset$ **faire**
  $t \leftarrow$ *oter_action*(*actions_obs_à_traiter*(q)) ; *explorées*$(q) \leftarrow$ *explorées*$(q) \cup \{t\}$
  Soit $q' | q \xrightarrow{t} q' \in E$; *statut*$(q) \leftarrow$ *fixé* ; *atteintes* $\leftarrow$ *atteintes* $\cup \{q \xrightarrow{t} q'\}$
  évitées2$(q') \leftarrow$ (*evitées*$(q) \cup$ *explorées*$(q)) \setminus$ (*incertaines*$(q) \cup$ *dépendantes*(t))
fin tant que
à_visiter$(q) \leftarrow \mathbf{faux}$
Sortie : *atteintes*

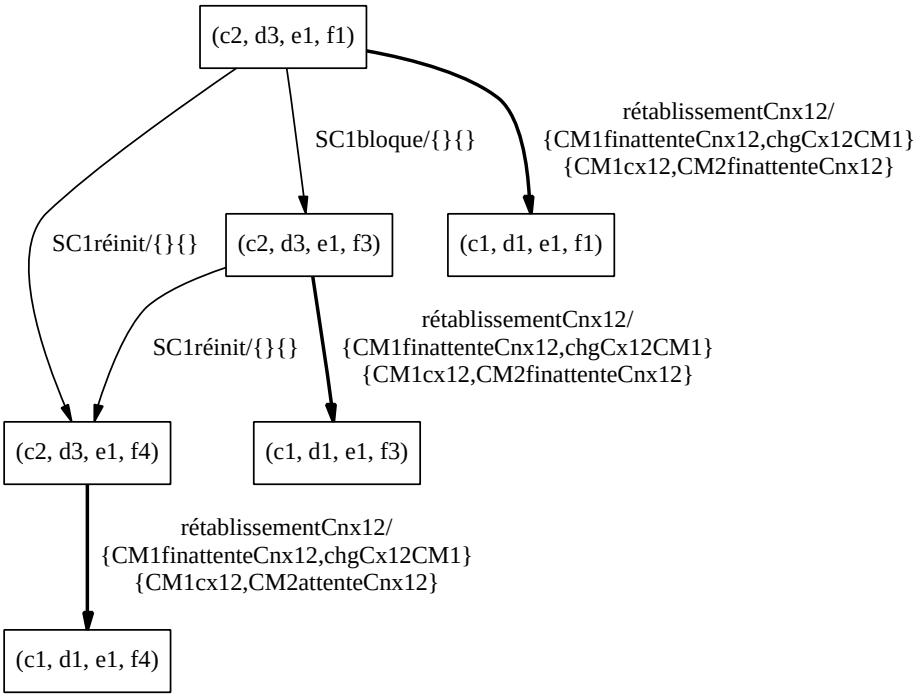

FIG. 4.3 – $DiagRed((c_2, d_3, e_1, f_1), \emptyset, CM1cx12)$.

Remarque 4.1 *Contrairement à* $DiagRed(q_0, e, o)$, $DiagRed(q_0, e)$ *a toujours une solution (qui peut être vide).* □

4.3.3 Algorithme en ligne

Nous présentons dans cette section un premier algorithme en ligne pour le calcul de $\Delta_\gamma(\mathcal{O}_\gamma)$.

On considère une nouvelle observation o issue de γ. Étant donnée l'hypothèse sur l'ordre total des observations sur γ, cette observation o ne peut être expliquée que par des comportements qui se déroulent après les états expliquant $\mathcal{O}_\gamma$.

L'adaptation du diagnostic local est présentée dans l'algorithme 3. L'idée est la suivante : à un instant donné, on dispose du diagnostic $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$; il recense les comportements de γ qui émettent l'ensemble d'observations $\mathcal{O}_\gamma$. Une hypothèse de diagnostic de $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ est constituée d'une partie résumant le comportement de γ avant l'émission et pendant l'émission des observations associées à $\mathcal{O}_\gamma$ et d'une partie résumant le comportement de γ après l'émission de toutes les observations associées à $\mathcal{O}_\gamma$. C'est cette partie de comportement qui est à remettre en question du fait de l'apparition de o. Dans un premier temps, on élimine les transitions associées à ces comportements (lignes 3-4). Puis, à partir des états expliquant $\mathcal{O}_\gamma$, on calcule les traces possibles expliquant o. À tout état $X = (q, \mathcal{O}_\gamma)$ est associé son ensemble *évitées(X)* (établi par d'anciens appels à $DiagRed$) qui est pris en compte lors du

calcul des traces. S'il existe au moins une trace (ligne 7), on en déduit les états et les transitions à insérer dans le nouveau diagnostic local et on met à jour les ensembles *évitées* (établis par l'algorithme 1) (lignes 11-15). Dans le cas contraire, aucune explication de o n'est possible à partir de l'état considéré, cet état est donc invalide et tout chemin de transitions y menant doit être éliminé (ligne 8). Cette élimination est nécessaire si l'on veut optimiser l'espace mémoire du diagnostic local, par contre, pour des fins d'efficacité temporelle, cette élimination peut être évitée [3].

La deuxième phase de l'algorithme consiste à établir le nouvel ensemble de comportements pouvant avoir lieu après l'observation de o. Cet ensemble est calculé à l'aide de l'ensemble de transitions $DiagRed(q, \textit{évitées}(X))$ où $X = (q, \mathcal{O}'_\gamma)$.

Initialisation du diagnostic La phase d'initialisation du diagnostic consiste à établir le diagnostic $\Delta_\gamma(\emptyset)$. Sans connaissance *a priori* sur un ensemble d'états possibles du système, l'initialisation consiste à établir l'ensemble des comportements non-observables de γ. Cet ensemble peut être établi en considérant l'ensemble des états de $\|\gamma\| = (I, O, Q, E)$ et en construisant :

$$\Delta^{red}_\gamma(\emptyset) = (I, O, Q', E')$$

où

$$E' = \{(q_1, \emptyset) \xrightarrow{t} (q_2, \emptyset) | q_1 \xrightarrow{t} q_2 \in DiagRed(q, \emptyset) \forall q \in Q\}$$

et

$$Q' = \{q, q' | q \xrightarrow{t} q' \in E'\}.$$

En pratique, on peut avoir une connaissance *a priori* sur un ensemble d'états possibles Q_0 de γ. Dans ce cas, on peut construire le diagnostic $\Delta^{red}(Q_0, \emptyset)$ en considérant non plus tous les états de γ mais uniquement ceux de Q_0. Cette connaissance *a priori* permet d'établir des diagnostics du type $\Delta^{red}(Q_0, \mathcal{O}_\gamma)$: les ensembles de comportements issus de Q_0 expliquant $\mathcal{O}_\gamma$. Bien évidemment, on a $\Delta^{red}(Q_0, \mathcal{O}_\gamma) \subseteq \Delta^{red}_\gamma(\mathcal{O}_\gamma)$.

4.3.4 Utilisation d'un diagnostiqueur

L'algorithme en ligne effectue des recherches en profondeur afin d'établir des traces servant d'explications aux observations reçues. Nous proposons d'éviter une partie de ce calcul lors du diagnostic en ligne en pré-compilant des informations nécessaires au diagnostic dans une structure appelée *diagnostiqueur local*. Cette structure notée $\mathcal{D}_\gamma$ pour le comportement local γ est un automate construit hors-ligne qui est en mesure de construire efficacement un diagnostic local en ligne étant donné un état initial du composant et une séquence d'observations. Un *diagnostiqueur local* est une adaptation des diagnostiqueurs fondés sur un modèle global qui sont proposés dans [Sampath et al. 95] et [Debouk et al. 98].

[3] On peut utiliser le principe du « ramasse-miette » (*garbage collector*) utilisé dans certains langages de programmation qui ne nettoie que lorsqu'il en a le temps.

Algorithme 3 Algorithme en ligne du diagnostic local

Entrée 1 : $\Delta_\gamma^{red}(\mathcal{O}_\gamma) = (I, O, Q, E)$

Entrée 2 : $o \in \Sigma_{obs}^\gamma$ {On note $\mathcal{O}'_\gamma$ l'ensemble des observations de $\mathcal{O}_\gamma$ suivies de o.}

$Q' \leftarrow Q \setminus \{(q, \mathcal{O}_\gamma) \in Q | (q', \mathcal{O}_\gamma) \xrightarrow{t} (q, \mathcal{O}_\gamma) \in E \wedge q \neq q' \wedge OBS_\gamma(q' \xrightarrow{t} q) = \emptyset\}$

$E' \leftarrow E \setminus \{X \xrightarrow{t} X' \in E | X \in Q', X' \in Q'\}$; $E'' \leftarrow \emptyset$; $Q'' \leftarrow \emptyset$

{Calcul du comportement expliquant l'observation o à partir des états de $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ où $\mathcal{O}_\gamma$ est déjà expliqué}

pour tout $X = (q, \mathcal{O}_\gamma) \in Q'$ **faire**

  si $DiagRed(q, \textit{évitées}(X), o) = \emptyset$ **alors**

    $(Q', E') \leftarrow \textit{éliminer\_transitions}(X, Q', E')$ {À partir de X, aucune explication de o n'est possible : élimination des transitions et des états menant uniquement à X}

  sinon

    pour tout $q_1 \xrightarrow{t} q_2 \in DiagRed(q, \textit{évitées(X)}, o)$ **faire**

      évitées$((q_1, \mathcal{O}_\gamma)) \leftarrow$ *évitées*(q_1) ;

      si $o \in OBS_\gamma(q_1 \xrightarrow{t} q_2)$ **alors**

        $Q'' \leftarrow Q'' \cup \{(q_1, \mathcal{O}_\gamma), (q_2, \mathcal{O}'_\gamma)\}$; $E'' \leftarrow E'' \cup \{(q_1, \mathcal{O}_\gamma) \xrightarrow{t} (q_2, \mathcal{O}'_\gamma)\}$

        évitées$((q_2, \mathcal{O}'_\gamma)) \leftarrow$ *évitées2*(q_2) ;

      sinon

        $Q'' \leftarrow Q'' \cup \{(q_1, \mathcal{O}_\gamma), (q_2, \mathcal{O}_\gamma)\}$; $E'' \leftarrow E'' \cup \{(q_1, \mathcal{O}_\gamma) \xrightarrow{t} (q_2, \mathcal{O}_\gamma)\}$

        évitées$((q_2, \mathcal{O}_\gamma)) \leftarrow$ *évitées*(q_2) ;

      fin si

    fin pour

  fin si

fin pour

{Calcul du comportement non-observable pouvant se produire après l'émission de o}

$Q''' \leftarrow \emptyset$

pour tout $X = (q, \mathcal{O}'_\gamma) \in Q''$ **faire**

  pour tout $q_1 \xrightarrow{t} q_2 \in DiagRed(q, \textit{évitées}(X))$ **faire**

    $Q''' \leftarrow Q''' \cup \{(q_1, \mathcal{O}'_\gamma), (q_2, \mathcal{O}'_\gamma)\}$; $E''' \leftarrow E''' \cup \{(q_1, \mathcal{O}'_\gamma) \xrightarrow{t} (q_2, \mathcal{O}'_\gamma)\}$

  fin pour

fin pour

Sortie : $\Delta_\gamma^{red}(\mathcal{O}'_\gamma) \leftarrow (I, O, Q' \cup Q'' \cup Q''', E' \cup E'' \cup E''')$

4.3.4.1 Observateur local

Le calcul du diagnostic local $\Delta_\gamma(\mathcal{O}_\gamma)$ est fondé sur la mise en place d'un *observateur* de γ [Cassandras et Lafortune 99]. L'observateur de γ, noté γ_{obs} est un automate déterministe à nombre fini d'états qui représente le comportement observable de γ, (voir définition 3.21 page 76).

Définition 4.11 (automate observateur) *Soit Q_0 un ensemble d'états de $\gamma = (I, O, Q, E)$, l'observateur γ_{obs} est un automate déterministe :*

$$(Q^{obs}, O, E^{obs}, q_0^{obs})$$

où

- *$Q^{obs} \subseteq 2^Q$ est l'ensemble des états de l'observateur ;*
- *$E^{obs} \subseteq Q^{obs} \times 2^O \times Q^{obs}$ est l'ensemble de transitions ;*
- *$q_0^{obs} \in Q^{obs}$ est l'état initial de l'observateur.*

Cet automate est obtenu à l'aide de l'algorithme 4. □

L'algorithme 4 présente la construction de l'observateur γ_{obs} à partir de γ et d'un ensemble d'états $Q_0 \subseteq Q$. L'ensemble Q_0 représente l'état initial de l'observateur, autrement dit, l'observateur fait l'hypothèse que l'état initial de γ est l'un des états de Q_0. Dans l'hypothèse où l'on n'a pas d'information sur l'état initial de γ, il faut alors construire γ_{obs} à partir de $Q_0 = Q$. Après l'initialisation de la structure servant de représentation pour γ_{obs} (*initialiser*(γ_{obs})), on construit l'état initial de l'observateur (*créer_état*(γ_{obs}, Q_0)). Pour chaque état construit q_{obs}, l'idée consiste à rechercher tous les comportements observables possibles. On recherche tous les chemins de transitions $q_1 \xrightarrow{t_1} q_2 \dots q_m \xrightarrow{t_m} q_{m+1}$ de γ tels que :

1. l'état source du chemin est un état de q_{obs} ($q_1 \in q_{obs}$) ;
2. les $m-1$ premières transitions ne sont pas observables ($\forall i \in \{1, \dots, m-1\}, OBS_\gamma(q_i \xrightarrow{t_i} q_{i+1} = \emptyset$) ;
3. la dernière transition émet l'ensemble observable $o = OBS_\gamma(q_m \xrightarrow{t_m} q_{m+1}) \neq \emptyset$.

Cette recherche est effectuée par *rechercher_états_accessibles* et produit l'ensemble des états accessibles par ces chemins (à savoir l'ensemble des états q_{m+1}). Cet ensemble constitue un nouvel état q'_{obs} de l'observateur, la transition munie de l'ensemble observable o entre q_{obs} et q'_{obs} est construite (*créer_transition*$(\gamma_{obs}, q_{obs}, o, q'_{obs})$). Par construction, cette transition exprime le fait qu'il existe au moins un chemin de transitions dans γ entre un état de q_{obs} et chaque état de q'_{obs} qui produit le comportement observable o [4]. On remarquera aussi que, par construction, tout état de γ contenu dans un état de l'observateur qui dispose d'une transition entrante [5] est un état cible d'une transition émettant des observables dans γ. Par la suite, de tels états seront appelés *états observables*.

[4] L'observateur ainsi construit dispose de transitions étiquetées par des ensembles d'observables ; dans la pratique, on essaiera d'établir un observateur sur un modèle $\|\gamma\|$ dont les comportements observables de chaque transition sont des singletons.

[5] Seul l'état initial de l'observateur peut ne pas en avoir.

Algorithme 4 Construction de l'observateur de γ.

Entrée 1 : $\gamma = (I, O, Q, E)$

Entrée 2 : Q_0 un ensemble d'états de γ

$OBS_\gamma \leftarrow \{OBS_\gamma(q \xrightarrow{t} q') | q \xrightarrow{t} q' \in E\}$

initialiser(γ_{obs})

états_à_traiter $\leftarrow \{$*créer_état*$(\gamma_{obs}, Q_0)\}$

états_traités $\leftarrow \emptyset$

tant que *états_à_traiter* $\neq \emptyset$ **faire**

  $q_{obs} \leftarrow$ *oter_état*(*états_à_traiter*)

  états_traités $\leftarrow$ *états_traités* $\cup \{q_{obs}\}$

  pour tout $o \in OBS_\gamma$ **faire**

    accessibles $\leftarrow$ *rechercher_états_accessibles*(γ, q_{obs}, o)

    si *accessibles* $\neq \emptyset$ **alors**

      $q'_{obs} \leftarrow$ *créer_état*$(\gamma_{obs},$ *accessibles*$)$

      créer_transition$(\gamma_{obs}, q_{obs}, o, q'_{obs})$

      si $q'_{obs} \notin$ *états_traités* **alors**

        états_à_traiter $\leftarrow$ *états_à_traiter* $\cup \{q'_{obs}\}$

      fin si

    fin si

  fin pour

fin tant que

Sortie : γ_{obs} l'automate observateur de γ.

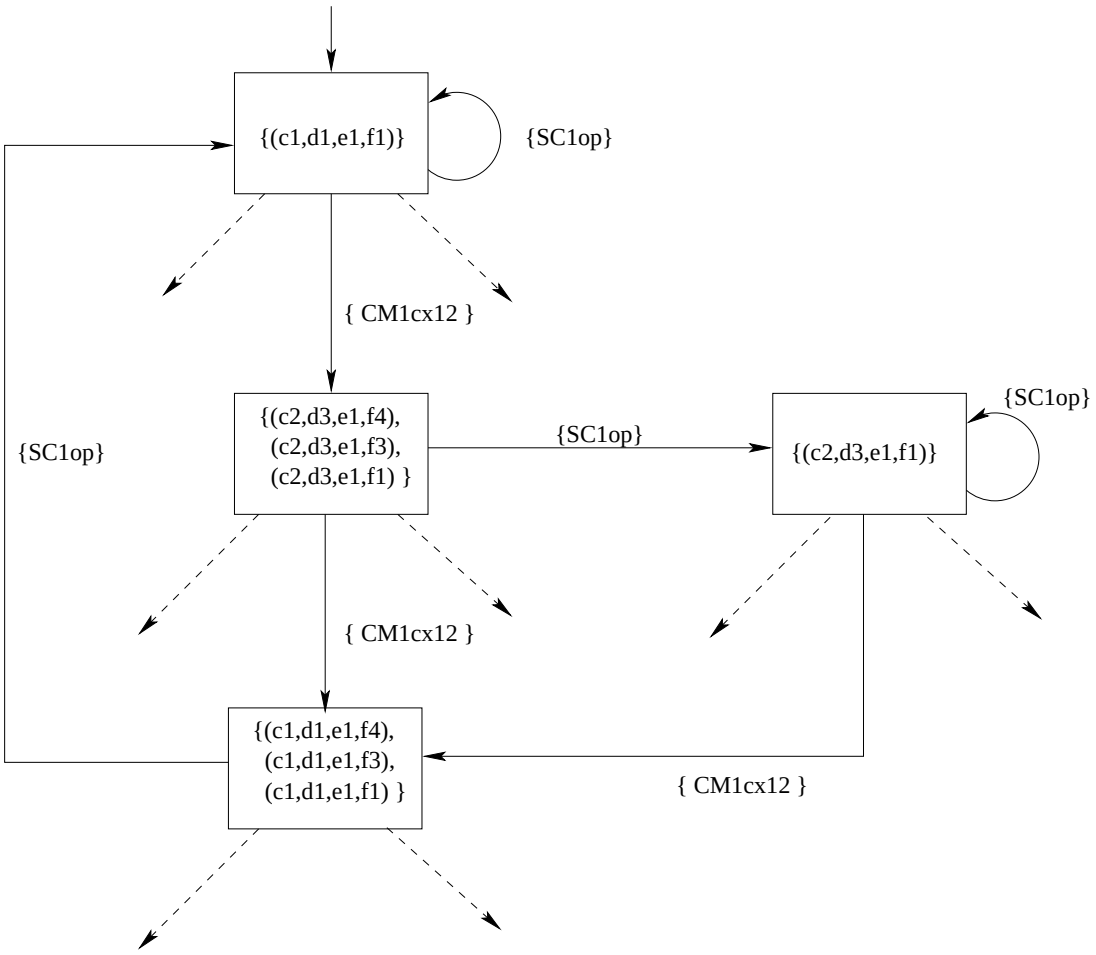

FIG. 4.4 – Partie de l'observateur de $\gamma = \{Cnx12, CM1ctl, CM1cnx, SC1\}$.

La figure 4.4 présente une partie de l'observateur de $\gamma = \{Cnx12, CM1ctl, CM1cnx, SC1\}$ (voir également la figure 4.1). Sur cette figure, on suppose que l'état initial est constitué uniquement de l'état (c_1, d_1, e_1, f_1) de $\|\gamma\|$. Parmi les comportements observables pouvant se produire sachant qu'on est dans l'état (c_1, d_1, e_1, f_1), on peut avoir l'émission de $\{SC1op\}$. Dans ce cas, il n'y a qu'un état cible : (c_1, d_1, e_1, f_1). Dans le cas de $\{CM1cx12\}$, les états cibles possibles sont $\{(c_2, d_3, e_1, f_4), (c_2, d_3, e_1, f_3), (c_2, d_3, e_1, f_1)\}$.

4.3.4.2 Diagnostiqueur local

Si on peut donner une définition informelle d'un diagnostiqueur alors il s'agit d'un *observateur renseigné*. L'observateur permet en effet de suivre le comportement observable de γ et d'informer sur les états possibles de γ après telle ou telle séquence d'observations. L'information que donne l'observateur est donc insuffisante car il ne nous donne pas les pannes que γ a pu subir. Le diagnostiqueur, tel que présenté dans [Sampath et al. 95, Sampath et al. 98, Debouk et al. 98, Debouk et al. 00a], est un observateur qui renseigne sur les pannes que peut subir γ en ajoutant dans chaque état de l'observateur des étiquettes de pannes, chaque étiquette représentant une hypothèse de panne possible. Ce diagnostiqueur est intéressant si l'on considère que γ n'interagit pas avec le reste du système. Or, dans le cas qui nous intéresse, γ interagit avec d'autres parties du système ; le diagnostic que le diagnostiqueur local doit fournir doit non seulement indiquer les pannes mais aussi les interactions possibles avec les composants voisins que ces pannes peuvent provoquer [Pencolé 00].

Étant donné un état de l'observateur, le diagnostiqueur local doit informer sur le comportement complet de γ expliquant l'arrivée dans cet état. Au lieu de donner de simples étiquettes de panne dans chaque état de l'observateur, l'information de diagnostic décrit un comportement local, un ensemble de chemins de transitions de γ. Deux types d'informations sont à distinguer.

Information 1 : que s'est-il passé pour que γ émette l'observation o ? Cette information est disponible à l'aide de la fonction $DiagRed(q, e, o)$ qui permet d'établir les traces de γ qui expliquent l'observation de o à partir d'un état q.

Information 2 : que se passe-t-il si γ n'émet plus d'observable ? Cette information est disponible à l'aide de la fonction $DiagRed(q, e)$ qui permet d'établir les traces de γ non observables issues de q.

Diagnostiqueur local, définition et construction La fonction $DiagRed$ construit les informations de diagnostic nécessaires à l'élaboration du diagnostic local de γ à l'aide de l'observateur. Afin d'augmenter l'efficacité du calcul du diagnostic local, on peut compiler ces informations. Le diagnostiqueur local est défini ainsi :

Définition 4.12 (Diagnostiqueur local) *Le* diagnostiqueur local *de γ, noté $\mathcal{D}_\gamma$, est le couple :*

$$\mathcal{D}_\gamma \triangleq (\gamma_{obs}^{red}, DiagRed)$$

où γ_{obs}^{red} est un observateur réduit de γ_{obs}. □

La construction du diagnostiqueur est fondé sur l'algorithme de construction de l'observateur (algorithme 4). L'idée est de profiter du calcul de l'observateur, pour précalculer $DiagRed(q, e, o)$ et $DiagRed(q, e)$. Ces précalculs peuvent être établis lors de la phase de recherche des états accessibles (fonction *recherche_états_accessibles* de l'algorithme 4). L'observateur construit n'est pas tout à fait γ_{obs}. En effet, chaque état de l'observateur construit recense uniquement les états accessibles par des chemins, chacun étant le représentant unique d'une trace et non pas tous les états accessibles (l'algorithme de construction de l'observateur γ_{obs}^{red} est identique à l'algorithme 4 à l'exception de l'appel à la fonction *rechercher_états_accessibles* qu'il faut remplacer par *rechercher_réduite_états_accessibles*). Tous les états non accédés par cette fonction mais accessibles par *rechercher_états_accessibles* sont équivalents, à des permutations d'événements indépendants près, à des états accessibles par la fonction de recherche réduite.

La figure 4.5 présente la partie du diagnostiqueur de $\gamma = \{Cnx12, CM1ctl, CM1cnx, SC1\}$ qui correspond à la partie de l'observateur de la figure 4.4. Chaque état de $\|\gamma\|$ contenu dans un état du diagnostiqueur est muni d'un ensemble de transitions pouvant être un comportement possible de $\|\gamma\|$ à partir de cet état et en fonction du comportement observable pouvant se produire. Dans cet exemple, l'observateur réduit γ_{obs}^{red} est identique à γ_{obs}.

4.3.4.3 Construction de diagnostics locaux

L'initialisation du diagnostic local est établie de la même façon que dans l'approche en ligne. Cette initialisation correspond de plus à l'initialisation de $\mathcal{D}_\gamma$: son état courant est l'état initial q_0^{obs} de l'observateur associé. Le diagnostic local $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ est représenté implicitement par la séquence de transitions de $\mathcal{D}_\gamma$ représentant $\mathcal{O}_\gamma$ [6]. Supposons que le diagnostic $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ est établi et qu'une nouvelle observation o est disponible. L'adaptation de $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ en $\Delta_\gamma^{red}(\mathcal{O}'_\gamma)$ consiste à considérer la transition de l'observateur γ_{obs}^{red} issue de l'état courant et étiquetée par o (une telle transition existe toujours du fait de l'hypothèse 3.4). Cette transition associe un ensemble source d'états de γ $\{q_1, \ldots, q_m\}$ à un ensemble cible d'états $\{q'_1, \ldots, q'_l\}$. L'ensemble source correspond dans le diagnostic local aux états courants, à savoir $\{(q_1, \mathcal{O}_\gamma), \ldots, (q_m, \mathcal{O}_\gamma)\}$, pour lesquels aucun comportement non-observable ne s'est produit après la dernière observation. La mise à jour du diagnostic local consiste alors à considérer les transitions $\bigcup_{i=1}^{m} DiagRed(q_i, \emptyset, o)$ afin de les insérer dans le nouveau diagnostic, puis pour chaque état de $\{q'_1, \ldots, q'_l\}$, on insère les transitions de $DiagRed(q'_i, \emptyset)$.

Il peut exister des états de $\{q_1, \ldots, q_m\}$ pour lesquels $DiagRed(q_i, \emptyset, o) = \emptyset$, cela signifie dans ce cas que o n'est pas explicable à partir de q_i. Les chemins de transitions aboutissant à $(q_i, \mathcal{O}_\gamma)$ dans $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ doivent donc être éliminés (application de *éliminer_transitions* de l'algorithme 3 (ligne 8)).

Réduction non-optimale du diagnostic La représentation obtenue par cette construction n'est pas réduite de façon optimale. Cela est dû à la construction du diagnostiqueur réduit

[6] Plus précisement, un chemin de transitions de $\mathcal{D}_\gamma$ représente un comportement observable de γ dont la jointure avec $\mathcal{O}_\gamma$ existe. Si des transitions sont étiquetées avec un ensemble de plusieurs observables, le chemin peut ne pas être unique. En pratique, on évitera une telle situation afin de garantir le déterminisme.

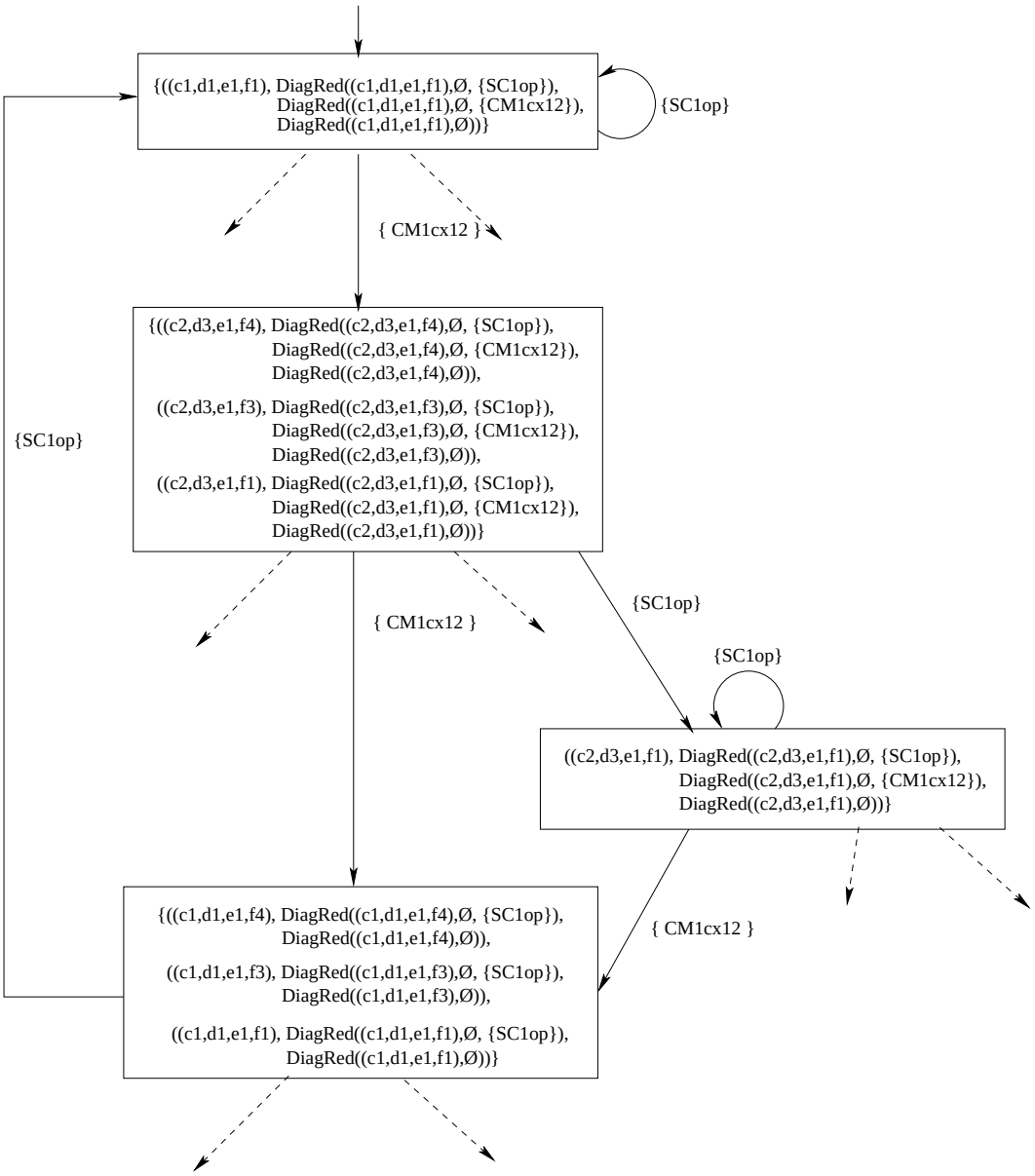

FIG. 4.5 – Partie du diagnostiqueur de $\gamma = \{Cnx12, CM1ctl, CM1cnx, SC1\}$.

qui n'est pas optimale. En effet, dans chaque état du diagnostiqueur, on précalcule les transitions issues de $DiagRed(q, \emptyset, o)$ et l'on ne considère jamais un ensemble initial *évitées*(q) non vide. Afin d'avoir une représentation plus réduite, il faut appliquer une nouvelle réduction. Pour chaque état $\{(q_1, \mathcal{O}_\gamma), \ldots, (q_m, \mathcal{O}_\gamma)\}$ de $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$, il suffit de conserver les ensembles *évitées*(q_i) issus de la dernière adaptation du diagnostic local. Au lieu d'ajouter dans le nouveau diagnostic local, toutes les transitions issues de $\bigcup_{i=1}^{m} DiagRed(q_i, \emptyset, o)$, on ajoute uniquement celles de

$$\bigcup_{i \in I} DiagRed(q_i, \textit{évitées}(q_i), o).$$

Par construction, on a en effet $DiagRed(q_i, \text{évitées}(q_i), o) \subseteq DiagRed(q_i, \emptyset, o)$. L'ensemble des états $q_i, i \in I$ où $I \subseteq \{1, \ldots, m\}$ est l'ensemble des états issus de $\Delta_\gamma^{red}(\mathcal{O}_\gamma)$ dans le cas où ce dernier est réduit de façon optimale : il se peut en effet que cet ensemble soit strictement inclus dans $\{q_1, \ldots, q_m\}$. L'ensemble des transitions à insérer dans le diagnostic local est établi en effectuant une nouvelle réduction sur l'ensemble des transitions $\bigcup_{i \in I} DiagRed(q_i, \emptyset, o)$ en considérant des ensembles *évitées*(q_i) non vides (si c'est le cas). L'ensemble des états de γ ainsi obtenus, cibles d'une transition étiquetée par o, est un sous-ensemble J de $\{q'_1, \ldots, q'_l\}$ représenté par un état de $\mathcal{D}_\gamma$, et à chacun de ces états, on associe son ensemble *évitées*(q'_i).

Pour la construction du comportement non-observable, on extrait de la même façon les transitions :

$$\bigcup_{i \in J} DiagRed(q'_i, \text{évitées}(q'_i)) \subseteq \bigcup_{i \in J} DiagRed(q'_i, \emptyset).$$

À chaque état courant de $\Delta_\gamma^{red}(\mathcal{O}'_\gamma)$, on associe son ensemble *évitées*(q'_i) issu de cette nouvelle réduction.

Que gagne-t-on en terme d'efficacité ? L'adaptation en ligne du diagnostic local présentée dans l'algorithme 3 nécessite le parcours en ligne du comportement de γ afin d'établir deux choses :

1. l'ensemble des comportements locaux expliquant la nouvelle observation ;
2. le calcul d'une réduction de ces comportements.

Grâce au diagnostiqueur, la plupart de ces parcours sont établis hors-ligne. La construction du diagnostic local consiste uniquement à insérer des transitions issues d'ensembles prédéfinis et contenus dans le diagnostiqueur. Il n'y a plus de parcours en ligne du comportement de γ, d'où un gain en efficacité. Rendre la réduction optimale nécessite d'appliquer une nouvelle réduction sur le diagnostic local obtenu mais cette réduction s'effectue sur un ensemble de transitions préréduit ce qui la rend plus efficace.

4.3.5 Optimisation du diagnostiqueur

Dans la méthode précédente, le diagnostic local réduit n'est pas optimal sans l'application en ligne d'une réduction supplémentaire. Ce problème est lié au fait que le diagnostiqueur n'est pas construit en tenant compte dans ses états des ensembles d'événements indépendants qu'on

peut éviter (*évitées*(q)). Cette section présente une nouvelle construction de diagnostiqueur $\mathcal{D}_\gamma^{opt}$ qui prend en compte ces ensembles. La propriété d'un tel diagnostiqueur est qu'il n'est plus nécessaire de faire des réductions en ligne pour obtenir un diagnostic local identique à celui qu'on peut obtenir par une construction en ligne complète.

Le calcul de $\mathcal{D}_\gamma^{opt}$ est identique à $\mathcal{D}_\gamma$, à l'exception de la fonction *rechercher_états_accessibles*(γ, q_{obs}, o). Cette fonction, au lieu de retourner un ensemble d'états de γ, retourne un ensemble de couples $(q,$ *évitées(q)*$)$. La différence avec le premier diagnostiqueur est que l'on considère que *évitées(q)* peut être non vide. Un état du diagnostiqueur $\mathcal{D}_\gamma^{opt}$ est donc un ensemble de couples de ce type. L'information associée à chaque état et donc précalculée hors-ligne est du type $DiagRed(q,$ *évitées(q)*$, o)$ et $DiagRed(q,$ *évitées(q)*$)$ où $(q,$ *évitées(q)*$)$ est un couple de l'état considéré de $\mathcal{D}_\gamma^{opt}$.

Algorithme 5 Construction d'un état du diagnostiqueur réduit de $\mathcal{D}_\gamma^{opt}$.

Fonction : *rechercher_états_accessibles*(γ, q_{obs}, o)

$atteintes \leftarrow \emptyset$

états_observés $\leftarrow \emptyset$

pour tout $(q,$ *évitées*$(q)) \in q_{obs}$ **faire**

atteintes $\leftarrow$ *atteintes* $\cup\, DiagRed(q,$ *évitées*$(q), o)$

fin pour

pour tout $q' \in$ *états_finals(atteintes)* **faire**

états_observés $\leftarrow$ *états_observés* $\cup \{(q',$ *évitées*$(q'))\}$

fin pour

Sortie : *états_observés*

La construction du diagnostic local à l'aide de $\mathcal{D}_\gamma^{opt}$ est identique à celle fondée sur l'utilisation de $\mathcal{D}_\gamma$ sans la phase de réduction en ligne.

Difficultés liées à $\mathcal{D}_\gamma^{opt}$ Une nouvelle information est insérée dans les états de $\mathcal{D}_\gamma^{opt}$, ce qui peut conduire à une explosion combinatoire du diagnostiqueur. Étant donnée la nature localisée de la construction du diagnostic *via* un diagnostiqueur, l'opération de réduction a un faible impact sur l'efficacité temporelle. Aussi, le gain de l'efficacité temporelle que $\mathcal{D}_\gamma^{opt}$ apporte, peut s'avérer négligeable face à l'augmentation de la complexité spatiale qu'il engendre.

4.3.6 Bilan

4.3.6.1 Comparatif des différentes approches

Cette section a permis de décrire un ensemble de méthodes pour la construction d'un diagnostic local. La construction en ligne consiste en un parcours en ligne du comportement de γ afin de détecter les comportements expliquant une observation et ne retenir qu'un ensemble restreint représentant l'ensemble. Cette technique fonctionne bien si le comportement de γ

à explorer n'est pas trop grand. Cette technique montre aussi que beaucoup d'informations calculées sont redondantes, aussi, afin d'optimiser l'efficacité de la construction en ligne du diagnostic local, l'idée consiste à précalculer des « bouts de comportements » dans un diagnostiqueur. En ligne, le traitement s'avère plus efficace, le diagnostiqueur est en mesure de suivre le comportement observé de γ et de fournir les comportements élémentaires afin d'établir le diagnostic local. Deux types de diagnostiqueurs sont envisagés. Le premier nécessite une phase de réduction en ligne du diagnostic local afin d'en avoir une représentation optimale. La construction d'un tel diagnostiqueur est fondée sur la construction classique d'un observateur. Le deuxième diagnostiqueur est optimal quant à l'utilisation en ligne, par contre, sa construction hors-ligne peut s'avérer trop complexe en taille mémoire, si bien que le gain obtenu en ligne devient guère intéressant.

4.3.6.2 Ordre partiel sur les observations locales

Tout au long de cette section, la construction du diagnostic local a été établie sous l'hypothèse 4.1. Si on lève cette hypothèse, alors les observations $\mathcal{O}_\gamma$ constituent un ordre partiel. Le problème de l'hypothèse d'ordre partiel sur les observations locales est que l'apparition d'une nouvelle observation remet en cause non plus uniquement le comportement non-observable diagnostiqué et se produisant après la dernière observation reçue mais tous les comportements qui expliquent des observations qui peuvent avoir été émises après celle qui vient d'être reçue. Cette remise en cause a des conséquences majeures sur l'efficacité de la construction du diagnostic local, notamment si on utilise une approche en ligne complète. Dans le cadre d'une approche de type diagnostiqueur, gérer l'ordre partiel revient à gérer l'ensemble des chemins de transitions du diagnostiqueur, chaque chemin représentant un comportement observable compatible avec l'ordre partiel des observations locales (la jointure des deux ensembles existe, voir définition 3.12 page 64). Gérer un ordre partiel d'observations avec une approche diagnostiqueur est possible en ligne. En effet, dès lors que l'on cherche à suivre le comportement observé de γ, il suffit de suivre le diagnostiqueur en fonction des observations. Si l'on ne cherche pas à consulter tous les comportements diagnostiqués après chaque réception d'observations, alors il est possible dans une approche diagnostiqueur d'attendre de mettre à jour le diagnostic local.

4.3.6.3 Plan de décentralisation : causes et conséquences

Le diagnostic local est fondé sur l'exploration réduite du comportement de γ, d'où une première conséquence : si γ est un composant élémentaire Γ_i, aucune réduction n'est possible, tout événement associé à Γ_i est dépendant de tout autre événement associé à ce même composant (voir la définition de la relation de dépendance D_γ (définition 4.9 page 86)).

L'efficacité de l'élaboration du diagnostic local dépend de plusieurs paramètres. Le premier est la taille du comportement de γ, si elle est assez importante, la construction en ligne du diagnostic local peut s'avérer inefficace. Ainsi, si dans le plan de décentralisation choisi, γ regroupe un ensemble important de composants élémentaires alors le comportement $\|\gamma\|$ est potentiellement grand, une approche de type diagnostiqueur est à envisager, sinon une approche en ligne est suffisante. Le deuxième facteur est la propriété sur l'ordre des observations. Si les

observations de $\|\gamma\|$ sont telles qu'il y a peu de relations entre les différentes observations, le calcul du diagnostic local est complexe, il est donc préférable de privilégier un plan de décentralisation dans lequel les observations potentiellement émises par les composants de γ sont munies d'une forte relation d'ordre : typiquement, c'est le cas dès lors que γ regroupe des composants élémentaires topologiquement proches.

4.4 Diagnostic global

4.4.1 Généralités

Le problème du calcul du diagnostic global a été subdivisé en un ensemble de calculs de diagnostics locaux fondés sur une décentralisation du modèle. Le problème restant à résoudre est la mise en place de la fusion des résultats afin d'établir le diagnostic du système. Dans le chapitre 3, nous avons vu que le calcul du diagnostic global consistait à *composer* les chemins de transitions afin de construire les comportemements globaux expliquant les observations (voir corollaire 3.1). Cette composition est fondée sur la synchronisation des transitions locales en vue d'établir des transitions synchronisées (au sens de la définition 3.6 page 61). Une difficulté est l'ordre partiel des observations qu'il faut prendre en compte si de telles relations existent.

Le deuxième point important concernant la fusion est qu'elle doit être la plus efficace possible. Pour cela, il faut tout d'abord que cette opération de fusion tire parti des diagnostics locaux réduits afin d'établir elle-même un diagnostic global réduit $\Delta^{red}(\mathcal{O})$. Cette fusion est fondée sur une propriété particulière de la relation de dépendance D_γ (voir définition 4.9 page 86).

Propriété 4.2 *Soit γ_1, γ_2 deux ensembles disjoints de composants élémentaires, soit $t_1^1, t_2^1 \in \mathcal{A}_{\gamma_1}$ deux actions de $\|\gamma_1\|$, si $(t_1^1, t_2^1) \notin D_{\gamma_1}$ alors on a :*

$$\forall t_1^2, t_2^2 \in \mathcal{A}_{\gamma_2}, ((t_1^1, t_1^2) \in \mathcal{A}_{\gamma_1 \cup \gamma_2} \wedge (t_2^1, t_2^2) \in \mathcal{A}_{\gamma_1 \cup \gamma_2}) \Rightarrow ((t_1^1, t_1^2), (t_2^1, t_2^2)) \notin D_{\gamma_1 \cup \gamma_2}.$$

□

Démonstration : Tout au long de cette démonstration, on considère deux actions $t_1^2, t_2^2 \in \mathcal{A}_{\gamma_2}$ telles que les produits (t_1^1, t_1^2) et (t_2^1, t_2^2) sont des labels issus de transitions localement synchronisées de $\|\gamma_1 \cup \gamma_2\|$, autrement dit $(t_1^1, t_1^2) \in \mathcal{A}_{\gamma_1 \cup \gamma_2} \subseteq \mathcal{A}_{\gamma_1} \times \mathcal{A}_{\gamma_2}$ et $(t_2^1, t_2^2) \in \mathcal{A}_{\gamma_1 \cup \gamma_2}$. Nous devons montrer que si $(t_1^1, t_2^1) \notin D_{\gamma_1}$, (t_1^1, t_1^2) et (t_2^1, t_2^2) vérifient les trois conditions de non-appartenance à $D_{\gamma_1 \cup \gamma_2}$ (voir définition 4.9 page 86).

1) Montrer que l'on a nécessairement $\mathcal{I}_{\Gamma \setminus \gamma_1 \cup \gamma_2}((t_1^1, t_1^2)) = \emptyset \vee \mathcal{I}_{\Gamma \setminus \gamma_1 \cup \gamma_2}((t_2^1, t_2^2)) = \emptyset$.

Si $(t_1^1, t_2^1) \notin D_{\gamma_1}$, d'après la définition 4.9 page 86, on a :

$$\mathcal{I}_{\Gamma \setminus \gamma_1}(t_1^1) = \emptyset \vee \mathcal{I}_{\Gamma \setminus \gamma_1}(t_2^1) = \emptyset.$$

Supposons que l'on a $\mathcal{I}_{\Gamma \setminus \gamma_1}(t_1^1) = \emptyset$, t_1^1 est donc le label d'une transition localement synchronisée de γ_1 (voir définition 3.17 page 70) qui n'émet pas d'événement vers des composants

de l'ensemble $\Gamma \setminus \gamma_1$, et pour les mêmes raisons elle n'en reçoit pas non plus. Toute transition de label t_1^1 est donc nécessairement activée par un événement exogène $e \in \Sigma_{exo}$ et n'émet que des événements internes à γ_1 ou des événements observables. Toute transition de label (t_1^1, t_1^2) est localement synchronisée et donc d'après la définition 3.17, t_1^2 ne peut être qu'un label issu d'un produit de transitions nulles (de label $\mathbf{e}|\{\}$, voir page 60). Par conséquent, $\mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_1^1, t_1^2)) = \emptyset$. De même, si l'on suppose que $\mathcal{I}_{\Gamma\setminus\gamma_1}(t_2^1) = \emptyset$, on montre également que $\mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_2^1, t_2^2)) = \emptyset$, d'où le résultat.

2) Montrer que l'on a nécessairement $\mathcal{I}_{\gamma_1\cup\gamma_2}((t_1^1, t_1^2)) \cap \mathcal{I}_{\gamma_1\cup\gamma_2}((t_2^1, t_2^2)) = \emptyset$.
On a vu dans le 1) qu'au moins l'un des deux labels t_1^2 ou t_2^2 ne peut être qu'un label issu d'un produit de transitions nulles. Supposons qu'il s'agisse de t_1^2, on a donc $\mathcal{I}_{\gamma_1\cup\gamma_2}((t_1^1, t_1^2)) = \mathcal{I}_{\gamma_1}(t_1^1)$ $(*)$.

$$\begin{aligned}\mathcal{I}_{\gamma_1\cup\gamma_2}((t_2^1, t_2^2)) &= \mathcal{I}_{\gamma_1\cup\gamma_2}(t_2^1) \cup \mathcal{I}_{\gamma_1\cup\gamma_2}(t_2^2) \text{ par définition} \\ &= \mathcal{I}_{\gamma_1}(t_2^1) \cup \mathcal{I}_{\gamma_2}(t_2^1) \cup \mathcal{I}_{\gamma_1}(t_2^2) \cup \mathcal{I}_{\gamma_2}(t_2^2)\end{aligned}$$

(t_2^1, t_2^2) est le label d'une transition localement synchronisée donc t_2^2 ne peut affecter par interaction dans γ_1 que des composants affectés par t_2^1, autrement dit $\mathcal{I}_{\gamma_1}(t_2^2) \subseteq \mathcal{I}_{\gamma_1}(t_2^1)$, d'où :

$$\mathcal{I}_{\gamma_1\cup\gamma_2}((t_2^1, t_2^2)) = \mathcal{I}_{\gamma_1}(t_2^1) \cup \mathcal{I}_{\gamma_2}(t_2^1) \cup \mathcal{I}_{\gamma_2}(t_2^2) \quad (**).$$

D'après $(*)$ et $(**)$ on a donc :

$$\begin{aligned}\mathcal{I}_{\gamma_1\cup\gamma_2}((t_1^1, t_1^2)) \cap \mathcal{I}_{\gamma_1\cup\gamma_2}((t_2^1, t_2^2)) &= \mathcal{I}_{\gamma_1}(t_1^1) \cap (\mathcal{I}_{\gamma_1}(t_2^1) \cup \mathcal{I}_{\gamma_2}(t_2^1) \cup \mathcal{I}_{\gamma_2}(t_2^2)) \\ &= \mathcal{I}_{\gamma_1}(t_1^1) \cap \mathcal{I}_{\gamma_1}(t_2^1) \text{ car } \gamma_1 \text{ et } \gamma_2 \text{ sont disjoints} \\ &= \emptyset \text{ car} (t_1^1, t_2^1) \notin D_{\gamma_1}.\end{aligned}$$

Pour les mêmes raisons, si l'on suppose que t_2^2 est un label issu d'un produit de transitions nulles, on aboutit au même résultat.

3) Montrer que l'on a nécessairement $(\mathcal{E}_{(t_1^1,t_1^2)} \cap \Sigma_{obs} = \emptyset \wedge \mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_1^1, t_1^2)) = \emptyset) \vee (\mathcal{E}_{(t_2^1,t_2^2)} \cap \Sigma_{obs} = \emptyset \wedge \mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_2^1, t_2^2)) = \emptyset) \vee (\mathcal{E}_{(t_1^1,t_1^2)} \cap \Sigma_{obs} = \mathcal{E}_{(t_2^1,t_2^2)} \cap \Sigma_{obs} \wedge \mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_1^1, t_1^2)) \cup \mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_2^1, t_2^2)) = \emptyset)$.

Comme $(t_1^1, t_2^1) \notin D_{\gamma_1}$, on distingue trois cas.

1. $\mathcal{E}_{t_1^1} \cap \Sigma_{obs} = \emptyset \wedge \mathcal{I}_{\Gamma\setminus\gamma_1}(t_1^1) = \emptyset$ est vrai. Dans ce cas, t_1^2 est issu de transitions nulles puisque $\mathcal{I}_{\Gamma\setminus\gamma_1}(t_1^1) = \emptyset$ et $(t_1^1, t_1^2) \in \mathcal{A}_{\gamma_1\cup\gamma_2}$. Par conséquent, on a $\mathcal{E}_{(t_1^1,t_1^2)} \cap \Sigma_{obs} = \emptyset \wedge \mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_1^1, t_1^2)) = \emptyset$.
2. $\mathcal{E}_{t_2^1} \cap \Sigma_{obs} = \emptyset \wedge \mathcal{I}_{\Gamma\setminus\gamma_1}(t_2^1) = \emptyset$ est vrai. Dans ce cas, t_2^2 est issu de transitions nulles puisque $\mathcal{I}_{\Gamma\setminus\gamma_1}(t_2^1) = \emptyset$ et $(t_2^1, t_2^2) \in \mathcal{A}_{\gamma_1\cup\gamma_2}$. Par conséquent, on a $\mathcal{E}_{(t_2^1,t_2^2)} \cap \Sigma_{obs} = \emptyset \wedge \mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_2^1, t_2^2)) = \emptyset$.

3. $\mathcal{E}_{t_1^1} \cap \Sigma_{obs} = \mathcal{E}_{t_2^1} \cap \Sigma_{obs} \wedge \mathcal{I}_{\Gamma\setminus\gamma_1}(t_1^1) \cup \mathcal{I}_{\Gamma\setminus\gamma_1}(t_2^1) = \emptyset$ est vrai. Dans ce cas, t_1^2 et t_2^2 sont tous les deux issus de transitions nulles, on a donc $\mathcal{E}_{(t_1^1,t_1^2)} \cap \Sigma_{obs} = \mathcal{E}_{(t_2^1,t_2^2)} \cap \Sigma_{obs} \wedge \mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_1^1,t_1^2)) \cup \mathcal{I}_{\Gamma\setminus\gamma_1\cup\gamma_2}((t_2^1,t_2^2)) = \emptyset$.

□

Cette propriété exprime le fait que si dans $\|\gamma_1\|$, on a détecté des actions indépendantes au sens de D_{γ_1}, cette indépendance n'est pas remise en question dans $\|\gamma_1 \cup \gamma_2\|$ au sens de $D_{\gamma_1\cup\gamma_2}$. Autrement dit, si l'on dispose de deux diagnostics réduits $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et $\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$, on sait que toute trace représentée par $\Delta^{red}_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2})$ est issue du produit libre de $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et $\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$.

4.4.2 Fusion de diagnostics

Cette section est consacrée à la description de l'opération de fusion entre deux diagnostics réduits $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et $\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$ afin d'établir le diagnostic réduit $\Delta^{red}_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2})$.

D'après les propriétés 3.5 et 4.2, une opération de fusion des diagnostics pourrait être la suivante :

1. calcul du produit libre $\langle\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}), \Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})\rangle$ en ne retenant que les transitions localement synchronisées par rapport à $\gamma_1 \cup \gamma_2$;
2. élimination des traces du résultat dont le comportement observable n'est pas compatible avec le comportement observé (de telles traces peuvent exister, du fait que la relation d'ordre de $\mathcal{O}_{\gamma_1\cup\gamma 2}$ est au moins plus stricte que celle définie par l'union des relations de $\mathcal{O}_{\gamma_1}$ et de $\mathcal{O}_{\gamma_2}$ sur ce même ensemble).

Le problème de cette opération est que le résultat obtenu n'est pas réduit de façon optimale (ce n'est pas $\Delta^{red}_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2})$), ce qui est très gênant du point de vue de l'efficacité. Cette perte d'optimalité vient du fait que l'opération produit ne prend pas en compte les caractères indépendants au sens de $D_{\gamma_1\cup\gamma_2}$ de certaines actions composées issues de $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et $\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$ (actions qui, avant composition, n'étaient pas considérées comme indépendantes) : il faut donc à nouveau procéder à une réduction, ce qui n'est guère efficace.

Une deuxième solution consiste à allier l'opération de composition classique de systèmes de transitions [Arnold 92] avec une exploration réduite [Peled 93] pour établir un unique représentant de chaque trace relative au diagnostic $\Delta_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2})$, à savoir $\Delta^{red}_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2})$. Cette opération notée $\odot$ est décrite par les algorithmes 6 et 7.

Définition de l'opération $\odot$

L'algorithme est fondé sur un parcours en profondeur et en parallèle des transducteurs $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et $\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$. Il s'agit d'un parcours « intelligent » qui détecte les chemins de transitions issus de $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et $\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$ dont la synchronisation explique les observations $\mathcal{O}_{\gamma_1\cup\gamma_2}$. Les chemins de transitions synchronisées retenus sont les uniques représentants de leur trace.

La fonction définie dans l'algorithme 6 est chargée d'effectuer ce parcours : elle suit le même principe que celui de $DiagRed$ et nécessite donc la gestion des mêmes ensembles

Algorithme 6 Composition et exploration réduite d'un état.

Fonction *visiter_état_composé*(q)

Entrée : $q = ((q_1, q_2), \mathcal{O}_{12}), \mathcal{O}_{12} = OBS(\mathcal{C}) \diamond \mathcal{O}_{12}^{\gamma_1 \cup \gamma_2}, \mathcal{O}_{12}^{\gamma_1 \cup \gamma_2} \sqsubseteq \mathcal{O}_{\gamma_1 \cup \gamma_2}, \mathcal{C}$ un chemin de transitions de $\|\gamma_1 \cup \gamma_2\|$ menant à (q_1, q_2)

état_visité$(q) \leftarrow \mathbf{vrai}$

actions_à_traiter$(q) \leftarrow \{t | q \xrightarrow{t=(t_1,t_2)} q', (q_1, P_{\gamma_1}(\mathcal{O}_{12})) \xrightarrow{t_1} (q_1', \mathcal{O}_1') \in E_1 \wedge (q_2, P_{\gamma_2}(\mathcal{O}_{12})) \xrightarrow{t_2} (q_2', \mathcal{O}_2') \in E_2 \wedge (q_1, q_2) \xrightarrow{t} (q_1', q_2') \in \|\gamma_1 \cup \gamma_2\| \wedge OBS(\mathcal{C}.(q_1, q_2) \xrightarrow{t} (q_1', q_2')) \diamond \mathcal{O}_{12}^{'\gamma_1 \cup \gamma_2}$ *existe*, $\mathcal{O}_{12}^{'\gamma_1 \cup \gamma_2} \sqsubseteq \mathcal{O}_{\gamma_1 \cup \gamma_2}\}$

tant que *actions_à_traiter*$(q) \neq \emptyset$ **faire**

  $t \leftarrow$ *oter_action*(*actions_à_traiter*(q)) ; *explorées*$(q) \leftarrow$ *explorées*$(q) \cup \{t\}$

  Soit $q' | q \xrightarrow{t} q'$

  à_éviter $\leftarrow$ (*evitées*$(q) \cup$ *explorées*$(q)) \setminus$ (*incertaines*$(q) \cup$ *dépendantes*(t))

  si $\neg$*état_visité*(q') **alors**

    explorées$(q') \leftarrow \emptyset$; *incertaines*$(q') \leftarrow \emptyset$; *évitées*$(q') \leftarrow$ *à_éviter*

    à_visiter$(q') \leftarrow \mathbf{vrai}$; *atteintes* $\leftarrow$ *atteintes* $\cup$ *visiter_état_composé*(q')

  sinon si *à_éviter* $\not\subseteq$ *évitées*(q') **alors**

    explorées$(q') \leftarrow \emptyset$; *incertaines*$(q') \leftarrow \emptyset$; *évitées*$(q') \leftarrow$ *évitées*$(q') \cap$ *à_éviter*

    si $\neg$*à_visiter*(q') **alors**

      à_visiter$(q') \leftarrow \mathbf{vrai}$; *atteintes* $\leftarrow$ *atteintes* $\cup$ *visiter_état_composé*(q')

    sinon

      actions_à_traiter$(q') \leftarrow$ *actions_à_traiter*$(q') \setminus$ *évitées*(q')

    fin si

  fin si

  si *statut*$(q') \in \{$*fixé*,*possible*$\} \vee$ *à_visiter*(q') **alors**

    atteintes $\leftarrow$ *atteintes* $\cup \{q \xrightarrow{t} q'\}$

    si *à_visiter*(q') **alors**

      incertaines$(q) \leftarrow$ *incertaines*$(q) \cup \{t\}$

    fin si

    si *statut*$(q') \in \{$*fixé*,*possible*$\}$ **alors**

      statut$(q) \leftarrow$ *statut*(q')

    sinon

      statut$(q) \leftarrow$ *possible* $\{q \xrightarrow{t} q'$ termine un cycle et q' est en cours de test.$\}$

    fin si

  fin si

fin tant que

si $\mathcal{O}_{12}^{'\gamma_1 \cup \gamma_2} = \mathcal{O}_{\gamma_1 \cup \gamma_2} \wedge \exists q \xrightarrow{t'} q' \in \|\gamma_1 \cup \gamma_2\|, OBS(q \xrightarrow{t'} q') \neq \emptyset$ ***alors***

  statut$(q) \leftarrow$ *fixé* $\{q$ *est un état final du diagnostic, il existe dans* $\|\gamma_1 \cup \gamma_2\|$ *un comportement observable issu de* q *qui pourrait expliquer des observations qui ne sont pas dans* $\mathcal{O}_{\gamma_1 \cup \gamma_2}$.$\}$

fin si

à_visiter$(q) \leftarrow \mathbf{faux}$

Sortie : *atteintes*

(*évitées, explorées,...,* voir page 93). Ce parcours est différent sur deux points. Le premier point est qu'il effectue l'exploration en créant les transitions et états explorés à la volée et le deuxième est qu'il détecte des traces expliquant l'ensemble des observations $\mathcal{O}_{\gamma_1 \cup \gamma_2}$ et non plus des traces menant à une unique observation. La fonction prend en paramètre un état $q = ((q_1, q_2), \mathcal{O}_{12})$. Cet état est le résultat de la composition d'un état $(q_1, \mathcal{O}_1)$ et d'un état $(q_2, \mathcal{O}_2)$ appartenant respectivement à $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1})$ et à $\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$. L'ensemble des observations $\mathcal{O}_{12}$ est l'ensemble jointure du comportement observable d'un chemin de transitions $\mathcal{C}$ menant à (q_1, q_2) dans $\|\gamma_1 \cup \gamma_2\|$ [7] et de l'ensemble préfixe $\mathcal{O}_{12}^{\gamma_1 \cup \gamma_2} \sqsubseteq \mathcal{O}_{\gamma_1 \cup \gamma_2}$ recensant ces observations. Si l'on note par $P_{\gamma_1}(\mathcal{O}_{12})$ l'ensemble partiellement ordonné induit de $\mathcal{O}_{12}$ et uniquement constitué des observations émises par les composants de γ_1, alors on a $P_{\gamma_1}(\mathcal{O}_{12}) = \mathcal{O}_1$ et $P_{\gamma_2}(\mathcal{O}_{12}) = \mathcal{O}_2$. Les actions à traiter issues de q sont les actions étiquetant les transitions localement synchronisées issues de q et qui sont compatibles avec l'ensemble des observations restant à expliquer à partir de q. Les transitions issues de q sont le résultat de la synchronisation des transitions issues de $(q_1, \mathcal{O}_1)$ et de $(q_2, \mathcal{O}_2)$. Par définition, ces actions font donc partie de celles issues de l'état (q_1, q_2) dans le comportement local $\|\gamma_1 \cup \gamma_2\|$.

On débute l'exploration de l'état q par l'une de ces actions (lignes 6-7). La fonction *oter_action* extrait une action selon un ordre particulier (l'ordre lexicographique par exemple), ce qui est nécessaire pour obtenir une représentation canonique du diagnostic (voir section 4.2.4 page 89). L'exploration réduite d'une action est identique à celle établie dans l'algorithme 2 page 95.

Si $\mathcal{O}_{12}^{\gamma_1 \cup \gamma_2} = \mathcal{O}_{\gamma_1 \cup \gamma_2}$, alors q est un état de diagnostic qui est la cible d'un comportement expliquant $\mathcal{O}_{\gamma_1 \cup \gamma_2}$, c'est un état final. Si (q_1, q_2) est source d'une transition observable dans $\|\gamma_1 \cup \gamma_2\|$, alors cela signifie que (q_1, q_2) pourrait expliquer des observations autres que celles de $\mathcal{O}_{\gamma_1 \cup \gamma_2}$, l'état q est donc fixé (lignes 32-34).

Le résultat de cette exploration est un ensemble d'états et transitions contenues dans $atteintes$. Cet ensemble contient un unique représentant pour chaque trace issue de q expliquant $\mathcal{O}_{\gamma_1 \cup \gamma_2}$ et tenant compte des éléments de *évitées*(q).

L'algorithme 7 définit l'opération $\odot$. Il initialise les différents parcours d'états. À partir de deux états initiaux $(q_1, \emptyset)$ et $(q_2, \emptyset)$ (ligne 8), on établit l'état produit $X_0 = ((q_1, q_2), \emptyset)$ (ligne 9). Si cet état n'a pas déjà été exploré, on l'explore à l'aide de la fonction définie par l'algorithme 6. Après l'exploration, il y a deux possibilités.

1. Le résultat de l'exploration contient au moins un état $((q'_1, q'_2), \mathcal{O}_{12})$, où $\mathcal{O}_{12}$ contient l'ensemble des observations de $\mathcal{O}_{\gamma_1 \cup \gamma_2}$, cela signifie dans ce cas qu'il existe au moins une trace d'événements expliquant $\mathcal{O}_{\gamma_1 \cup \gamma_2}$ issue de l'état q_0. Après une éventuelle élimination de transitions liée à l'exploration réduite (lignes 16-21), on conserve les chemins de transitions représentant toutes les traces expliquant $\mathcal{O}_{\gamma_1 \cup \gamma_2}$ issues de q_0 (ligne 22).
2. Le résultat ne contient pas d'état $((q'_1, q'_2), \mathcal{O}_{12})$, l'exploration a échoué, il n'existe pas à partir de X_0 de trace d'événements expliquant $\mathcal{O}_{\gamma_1 \cup \gamma_2}$. L'ensemble des transitions générées est ignoré.

L'ensemble des états et transitions retenus représente l'ensemble des traces d'événements expliquant les observations $\mathcal{O}_{\gamma_1 \cup \gamma_2}$ à partir de tous les états initiaux possibles de $\|\gamma_1 \cup \gamma_2\|$ selon

[7] Par construction, tout chemin de ce type a le même comportement observable.

les diagnostics de $\|\gamma_1\|$ et de $\|\gamma_2\|$. Ces ensembles constituent une représentation réduite du diagnostic de $\|\gamma_1 \cup \gamma_2\|$, à savoir $\Delta^{red}_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2})$.

Caractéristiques de l'opération $\odot$ Cette opération est fondée sur l'alliance entre une opération de composition et une opération de calcul réduit de traces. Cette opération construit les états et transitions composés à la volée lors de l'exploration réduite, ce qui la rend plus efficace qu'une opération constituée d'une phase de composition puis d'une phase de réduction. Par définition de l'opération $\odot$ on a la propriété suivante.

Propriété 4.3 *Soit γ_1 et γ_2 deux sous-ensembles distincts de Γ, soit $\mathcal{O}$ les observations du système, on a :*

$$\Delta^{red}_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2}) = \Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) \odot \Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$$

où $\forall\gamma \subseteq \Gamma, \mathcal{O}_\gamma = P_\gamma(\mathcal{O})$. □

Démonstration : La preuve de ce résultat est dans la définition même de l'opération $\odot$. □

Par définition, cette opération est commutative car elle est issue d'une composition. Une autre caractéristique intéresssante de cette opération est qu'elle est aussi associative.

Propriété 4.4 *Soit γ_1, γ_2 et γ_3 trois sous-ensembles distincts de Γ, soit $\mathcal{O}$ les observations du système, on a :*

$$\begin{aligned}\Delta^{red}_{\gamma_1\cup\gamma_2\cup\gamma_3}(\mathcal{O}_{\gamma_1\cup\gamma_2\cup\gamma_3}) &= (\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) \odot \Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})) \odot \Delta^{red}_{\gamma_3}(\mathcal{O}_{\gamma_3}) \\ &= \Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) \odot (\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2}) \odot \Delta^{red}_{\gamma_3}(\mathcal{O}_{\gamma_3}))\end{aligned}$$

où $\forall\gamma \subseteq \Gamma, \mathcal{O}_\gamma = P_\gamma(\mathcal{O})$. □

Démonstration : D'après la propriété 4.3, $\Delta^{red}_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2}) = \Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) \odot \Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$ et $\Delta^{red}_{\gamma_1\cup\gamma_2\cup\gamma 3}(\mathcal{O}_{\gamma_1\cup\gamma_2\cup\gamma_3}) = \Delta^{red}_{\gamma_1\cup\gamma_2}(\mathcal{O}_{\gamma_1\cup\gamma_2}) \odot \Delta^{red}_{\gamma_3}(\mathcal{O}_{\gamma_3})$ donc

$$\Delta^{red}_{\gamma_1\cup\gamma_2\cup\gamma_3}(\mathcal{O}_{\gamma_1\cup\gamma_2\cup\gamma_3}) = (\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) \odot \Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})) \odot \Delta^{red}_{\gamma_3}(\mathcal{O}_{\gamma_3}).$$

De même, $\Delta^{red}_{\gamma_2\cup\gamma_3}(\mathcal{O}_{\gamma_2\cup\gamma_3}) = \Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2}) \odot \Delta^{red}_{\gamma_3}(\mathcal{O}_{\gamma_3})$ et $\Delta^{red}_{\gamma_1\cup\gamma_2\cup\gamma 3}(\mathcal{O}_{\gamma_1\cup\gamma_2\cup\gamma_3}) = \Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) \odot \Delta^{red}_{\gamma_2\cup\gamma_3}(\mathcal{O}_{\gamma_2\cup\gamma_3})$ donc

$$\Delta^{red}_{\gamma_1\cup\gamma_2\cup\gamma_3}(\mathcal{O}_{\gamma_1\cup\gamma_2\cup\gamma_3}) = \Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) \odot (\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2}) \odot \Delta^{red}_{\gamma_3}(\mathcal{O}_{\gamma_3})).$$

□

La commutativité et l'associativité de $\odot$ sont des propriétés intéressantes car elles autorisent la fusion séparée de diagnostics locaux et dans n'importe quel ordre. Finalement, l'application de cette opération à l'ensemble des diagnostics locaux conduit à la production du diagnostic du système.

Algorithme 7 Fusion de diagnostics : $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) \odot \Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2})$

Entrée 1 : $\Delta^{red}_{\gamma_1}(\mathcal{O}_{\gamma_1}) = (I_1, O_1, Q_1, E_1)$
Entrée 2 : $\Delta^{red}_{\gamma_2}(\mathcal{O}_{\gamma_2}) = (I_2, O_2, Q_2, E_2)$
Entrée 3 : $\mathcal{O}_{\gamma_1 \cup \gamma_2}$ {La connaissance de $\mathcal{O}_{\gamma_1 \cup \gamma_2}$ est nécessaire uniquement s'il existe une relation d'ordre entre des observations de $\mathcal{O}_{\gamma_1}$ et de $\mathcal{O}_{\gamma_2}$}
$Q \leftarrow \emptyset$; $E \leftarrow \emptyset$
pour tout $X \in Q_1 \times Q_2$ **faire**
  statut$(X) \leftarrow$ *en_test*
fin pour
pour tout $X_1 = (q_1, \emptyset) \in Q_1, X_2 = (q_2, \emptyset) \in Q_2$ **faire**
  $X_0 \leftarrow ((q_1, q_2), \emptyset)$
  si *statut*$(X_0) =$ *en_test* **alors**
    $Q' \leftarrow \emptyset$; $E' \leftarrow \emptyset$
    explorées$(X_0) \leftarrow \emptyset$; *incertaines*$(X_0) \leftarrow \emptyset$; *à_visiter*$(X_0) \leftarrow \mathbf{vrai}$; *évitées*$(X_0) \leftarrow \emptyset$
    $(Q', E') \leftarrow$ *visiter_état_composé*(X_0)
    si $\exists X \in Q' | X = ((q'_1, q'_2), \mathcal{O}_{12}), |\mathcal{O}_{12}| = |\mathcal{O}_{\gamma_1 \cup \gamma_2}|$ **alors**
      {Il y a au moins une solution.}
      pour tout $X \xrightarrow{t} X' \in E' |$ *statut*$(X) =$ *possible* $\wedge$ *statut*$(X') =$ *fixé* **faire**
        statut$(X) \leftarrow$ *fixé* {Un état possible, source d'une transition dont la cible est fixée, est un état fixé}
      fin pour
      pour tout $X \xrightarrow{t} X' \in E' | \neg($*statut*$(X) =$ *fixé* $\wedge$ *statut*$(X') =$ *fixé*$)$ **faire**
        $E' \leftarrow E' \setminus \{X \xrightarrow{t} X'\}$ {Élimination des transitions dont un des états n'est pas fixé}
      fin pour
      $Q \leftarrow Q \cup Q'$; $E \leftarrow E \cup E'$
    fin si
  fin si
fin pour
Sortie : $\Delta^{red}_{\gamma_1 \cup \gamma_2}(\mathcal{O}_{\gamma_1 \cup \gamma_2}) = (I_1 \times I_2, O_1 \times O_2, Q, E)$

Propriété 4.5 *Soit $\{\gamma_1, \ldots, \gamma_m\}$ une décentralisation du système supervisé, soit $\mathcal{O}$ les observations du système, le diagnostic du système est obtenu par :*

$$\Delta^{red}(\mathcal{O}) = \bigodot_{i=1}^{m} \Delta^{red}_{\gamma_i}(\mathcal{O}_{\gamma_i}).$$

□

4.4.3 Conclusion

Cette section a présenté l'opération $\odot$. Cette opération est utilisée pour établir la fusion des diagnostics locaux en vue d'établir une représentation réduite du diagnostic global. Cette opération nécessaire est fondée sur une composition de systèmes de transitions couplée avec une exploration réduite afin d'être la plus efficace possible. Néanmoins, malgré cette optimisation, cette opération est sujette à des problèmes de complexité, car elle peut produire dans le pire des cas un transducteur qui n'est ni plus ni moins que le produit cartésien des diagnostics fusionnés.

La section suivante a pour objectif de présenter une stratégie pour la fusion des diagnostics qui permet d'optimiser encore le coût de la fusion. Elle permet de plus d'éviter au mieux l'application des fusions dans les pires cas, tout en ayant au final le diagnostic du système.

4.5 Stratégie de fusion

4.5.1 Constats

L'opération de fusion $\odot$ est une opération dont la complexité est liée au nombre d'états et de transitions à synchroniser. Si un diagnostic $\Delta^{red}_{\gamma_1}$ dispose de n_1 états et un diagnostic $\Delta^{red}_{\gamma_2}$ dispose de n_2, au pire l'espace de recherche de l'opération $\odot$ est de $n_1 \times n_2$ états.

Partant de ce constat, il est nécessaire d'utiliser l'opération de fusion avec parcimonie, en cherchant à l'éviter quand c'est possible, l'application de $\odot$ menant à des explorations au pire. Une stratégie de fusion est nécessaire si on veut espérer mettre en œuvre une plate-forme de diagnostic qui donne des résultats en ligne. Cette stratégie est constituée de plusieurs points et suit le principe du « moindre effort » [Pencolé et al. 02]. Pour alléger les notations, le diagnostic local de γ sera noté Δ_γ, il s'agit néanmoins de sa représentation réduite.

4.5.2 Élimination d'hypothèses locales impossibles

Principe

L'opération de fusion $\odot$ est une opération dont la complexité est liée au nombre d'états et de transitions à synchroniser. Aussi, afin de diminuer ce coût, une idée consiste à éliminer des diagnostics locaux un ensemble de transitions associées à des hypothèses locales que l'on sait déjà incompatibles suivant un critère simple à évaluer.

Ce critère porte sur les interactions qu'un diagnostic suppose. Une interaction est un échange d'événements entre deux composants élémentaires. Les interactions intéressantes sont celles qui font intervenir deux composants dont le comportement est décrit dans deux transducteurs $\|\gamma_i\|$ et $\|\gamma_j\|$ différents.

Calcul des interactions locales

On considère un comportement local $\|\gamma\| = \|\Gamma_{i_1}, \ldots, \Gamma_{i_k}\|$. Soit $\mathcal{C} = t_1, \ldots, t_m$ un chemin de transitions du diagnostic Δ_γ qui contient au moins une transition qui suppose l'échange de $e, e \in \Sigma_{int}$. On considère $t_j = (q_{i_1} \overset{e_{i_1}|\mathcal{I}_{i_1}\cup\mathcal{O}_{i_1}}{\longrightarrow} q'_{i_1}, \ldots, q_{i_k} \overset{e_{i_k}|\mathcal{I}_{i_k}\cup\mathcal{O}_{i_k}}{\longrightarrow} q'_{i_k})$ (pour les notations voir section 3.4.2.1 page 70). À partir de cet ensemble de transitions, on calcule les ensembles d'événements $\mathcal{E}_{t_j}(e)$ définis par :

$$\mathcal{E}_{t_j}(e) = \{\mathbf{e}\} \text{ si } ((\{e_{i_1}, \ldots, e_{i_k}\} \cup \bigcup_{j=1}^{k} \mathcal{O}_{i_j}) \setminus \{e\}) \cap \Sigma_{int} = \emptyset$$
$$= ((\{e_{i_1}, \ldots, e_{i_k}\} \cup \bigcup_{j=1}^{k} \mathcal{O}_{i_j}) \setminus \{e\}) \cap \Sigma_{int} \text{ sinon.}$$

On définit l'ensemble des événements échangés avec e lors du franchissement de ce chemin par l'ensemble d'événements $\mathcal{E}_{\mathcal{C}}(e)$ tel que :

$$\mathcal{E}_{\mathcal{C}}(e) \triangleq \begin{cases} \{\mathbf{e}\} & \text{si } \forall j \in \{1, \ldots, l\}, \mathcal{E}_{t_j}(e) = \{\mathbf{e}\}, \\ \bigcup_{j=1}^{l} \mathcal{E}_{t_j}(e) \setminus \{\mathbf{e}\} & \text{sinon.} \end{cases}$$

Un tel ensemble $\mathcal{E}_{\mathcal{C}}(e)$ signifie :
- si $\mathcal{E}_{\mathcal{C}}(e) = \{\mathbf{e}\}$ alors l'échange de e lors du franchissement du chemin $\mathcal{C}$ ne nécessite pas l'échange d'un autre événement ;
- si $\mathcal{E}_{\mathcal{C}}(e) \neq \{\mathbf{e}\}$ alors l'échange de e lors du franchissement du chemin $\mathcal{C}$ nécessite l'échange de tous les événements de $\mathcal{E}_{\mathcal{C}}(e)$.

Extension au diagnostic Le diagnostic Δ_γ est un ensemble de chemins de transitions issus de $\|\gamma\|$. On peut calculer l'ensemble des $\mathcal{E}_{\Delta_\gamma}(e)$ associés à ce diagnostic. Soit $\mathcal{C}_1, \ldots, \mathcal{C}_m$ l'ensemble de ces chemins. Si on considère ceux qui échangent l'événement e (notons-les $\mathcal{C}_{i_1}, \ldots, \mathcal{C}_{i_l}$), alors on associe au diagnostic Δ_γ l'ensemble $\mathcal{E}_{\Delta_\gamma}(e)$ tel que :

$$\mathcal{E}_{\Delta_\gamma}(e) \triangleq \begin{cases} \bigcup_{j=i_1}^{i_l} \mathcal{E}_{\mathcal{C}_j}(e) & \text{si } \forall j \in \{1, \ldots, l\}, \mathcal{E}_{\mathcal{C}_{i_j}}(e) \neq \{\mathbf{e}\} \\ \{\mathbf{e}\} & \text{sinon.} \end{cases}$$

Un tel ensemble $\mathcal{E}_{\Delta_\gamma}(e)$ signifie :
- si $\{\mathbf{e}\} = \mathcal{E}_{\Delta_\gamma}(e)$ alors le diagnostic dispose d'une hypothèse où e peut être échangé seul ;
- sinon pour tout événement f de $\mathcal{E}_{\Delta_\gamma}(e)$, il existe au moins une hypothèse de diagnostic qui suppose l'échange de f et de e.

Coordination des interactions fondées sur le diagnostic local

Une fois que ces interactions sont établies pour chaque diagnostic local, une coordination est établie en présence des interactions de tous les diagnostics locaux, c'est-à-dire, l'ensemble des événements échangés selon tous les diagnostics locaux. L'objectif de la coordination est de détecter les incohérences de point de vue sur ces échanges entre les différents diagnostics. Elle se fonde sur la propriété suivante.

Propriété 4.6 *Si* $\{\mathbf{e}\} \neq \mathcal{E}_{\Delta_\gamma}(e)$ *et si on sait qu'aucun événement de* $\mathcal{E}_{\Delta_\gamma}(e)$ *n'a pu être échangé, alors l'échange de* e *est impossible.* □

Démonstration : Si $\mathbf{e}$ n'est pas dans $\mathcal{E}_{\Delta_\gamma}(e)$, cela signifie qu'il existe au moins un événement de $\mathcal{E}_{\Delta_\gamma}(e)$ qui doit être échangé en même temps que e selon le diagnostic Δ_γ. Si l'on sait d'après les autres diagnostics locaux qu'aucun événement de $\mathcal{E}_{\Delta_\gamma}(e)$ n'a pu être échangé, cela signifie alors que les hypothèses de Δ_γ sur l'échange de e ne sont pas possibles. □

La coordination est effectuée à l'aide de l'algorithme 8 qui détecte en présence de toutes les interactions celles qui ne sont pas possibles. Si on considère un plan de décentralisation $\{\gamma_1, \ldots, \gamma_m\}$, l'ensemble $\mathcal{T}_i$ représente l'ensemble des couples $(e, \mathcal{E}_{\Delta_{\gamma_i}}(e))$ où e est un événement émis par $\|\gamma_i\|$ selon Δ_{γ_i} et l'ensemble $\mathcal{R}_i$ représente l'ensemble des couples $(e, \mathcal{E}_{\Delta_{\gamma_i}}(e))$ où e est un événement reçu par $\|\gamma_i\|$ selon Δ_{γ_i}. La première phase de coordination consiste à vérifier que tout événement e appartenant à un couple $(e, \mathcal{E}_{\Delta_{\gamma_i}}(e))$ des $\mathcal{T}_i$ est bien dans un couple $(e, \mathcal{E}_{\Delta_{\gamma_j}}(e))$ de $\mathcal{R}_j$ et que tout événement e appartenant à un couple $(e, \mathcal{E}_{\Delta_{\gamma_j}}(e))$ des $\mathcal{R}_j$ est bien dans un couple $(e, \mathcal{E}_{\Delta_{\gamma_i}}(e))$ de $\mathcal{T}_i$ (lignes 3-18). Tout événement ne répondant pas à ce critère n'a pu être effectivement échangé ; l'interaction est impossible.

Une fois ces interactions impossibles détectées, on répercute cette impossibilité sur les autres (lignes 20-29). On considère un événement impossible e et on l'élimine des ensembles $\mathcal{E}_{\Delta_{\gamma_k}}(e')$ de tous les couples connus. Si cette élimination conduit à ce que $\mathcal{E}_{\Delta_{\gamma_k}}(e')$ devienne vide, alors cela signifie, d'après la propriété 4.6, que l'événement e' ne peut pas être échangé non plus. Cette nouvelle information doit être répercutée.

Le résultat de cet algorithme est un ensemble d'événements supposés échangés par certains diagnostics locaux mais qui ne peuvent pas l'être selon les autres. Toute hypothèse de diagnostic qui suppose l'échange de l'un de ces événements est impossible.

Définition 4.13 (Hypothèse impossible) *Une* hypothèse impossible *du diagnostic local* Δ_γ *est un chemin de transitions dans lequel il existe une transition qui suppose l'échange d'un événement* e *issu de l'algorithme 8.* □

Nous appelons *diagnostic épuré* de γ, l'ensemble des hypothèses de Δ_γ possibles. D'après l'hypothèse 3.4, un tel diagnostic existe toujours, nous le noterons dans la suite de cette section par Δ'_γ.

Algorithme 8 Algorithme de calcul des interactions impossibles

Entrée : $\mathcal{T}_1, \ldots, \mathcal{T}_m, \mathcal{R}_1, \ldots, \mathcal{R}_m$ {Événements transmis et reçus.}
$\grave{a}\_répercuter \leftarrow \emptyset$; $\grave{a}\_éliminer \leftarrow \emptyset$
{Détection des interactions impossibles selon $\mathcal{T}_1 \ldots \mathcal{T}_m$.}
pour tout $i \in \{1, \ldots, m\}$ **faire**
  pour tout $(e, \mathcal{E}_{\Delta_{\gamma_i}}(e)) \in \mathcal{T}_i$ **faire**
    si $\nexists k \in \{1, \ldots, m\} \setminus \{i\}, (e, \mathcal{E}_{\Delta_{\gamma_k}}(e)) \in \mathcal{R}_k$ **alors**
      $\mathcal{E}_{\Delta_{\gamma_i}}(e) \leftarrow \emptyset$; $inserer\_queue(\grave{a}\_répercuter, e)$ {L'émission de e est impossible.}
    fin si
  fin pour
fin pour
{Détection des interactions impossibles selon $\mathcal{R}_1 \ldots \mathcal{R}_m$.}
pour tout $i \in \{1, \ldots, m\}$ **faire**
  pour tout $(e, \mathcal{E}_{\Delta_{\gamma_i}}(e)) \in \mathcal{R}_i$ **faire**
    si $\nexists k \in \{1, \ldots, m\} \setminus \{i\}, (e, \mathcal{E}_{\Delta_{\gamma_k}}(e)) \in \mathcal{T}_k$ **alors**
      $\mathcal{E}_{\Delta_{\gamma_i}}(e) \leftarrow \emptyset$; $inserer\_queue(\grave{a}\_répercuter, e)$ {La réception de e est impossible.}
    fin si
  fin pour
fin pour
{Calcul des répercussions des événements incompatibles sur les autres.}
tant que $\grave{a}\_répercuter \neq \emptyset$ **faire**
  $e \leftarrow oter\_tête(\grave{a}\_répercuter)$
  pour tout $(e', \mathcal{E}_{\Delta_{\gamma_k}}(e')) \in \mathcal{R}_1 \cup \cdots \cup \mathcal{R}_m \cup \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_m | e \in \mathcal{E}_{\Delta_{\gamma_k}}(e')$ **faire**
    $\mathcal{E}_{\Delta_{\gamma_k}}(e') \leftarrow \mathcal{E}_{\Delta_{\gamma_k}}(e') \setminus \{e\}$
    si $\mathcal{E}_{\Delta_{\gamma_k}}(e') = \emptyset$ **alors**
      $inserer\_queue(\grave{a}\_répercuter, e')$ {L'échange de e' est impossible.}
    fin si
  fin pour
  $\grave{a}\_éliminer \leftarrow \grave{a}\_éliminer \cup \{e\}$
fin tant que
Sortie : $\grave{a}\_éliminer$ {L'ensemble des événements qui rendent impossibles certaines hypothèses locales de diagnostic.}

4.5.3 Planifications des fusions

L'objectif de la fusion est de vérifier qu'une hypothèse proposée par un diagnostic local est valide en la confrontant à celles des autres diagnostics locaux. Une hypothèse locale est valide si les interactions qu'elle suppose sont possibles du point de vue des autres diagnostics. Par conséquent, il est facile de voir que fusionner des hypothèses qui ne supposent aucune interaction ne sert à rien, car elles sont *a fortiori* valides.

Ce constat nous invite à mettre au point un plan des fusions de diagnostics qui sont utiles et qui évite des fusions qui n'apportent plus aucune information de diagnostic. Un tel plan détermine la manière dont on va appliquer la fusion des diagnostics pour obtenir le résultat final : tout plan est possible étant donné que l'opération $\odot$ est commutative et associative (voir propriété 4.4).

Fusions à privilégier

Notation : on note $\mathcal{I}_{\Delta_{\gamma_i}}(\gamma_j)$ l'ensemble des événements des composants de γ_i qui sont supposés avoir été échangés avec les composants de γ_j d'après le diagnostic local de γ_i (à savoir Δ_{γ_i}).

Définition 4.14 (Grappes interagissantes) *γ_i et γ_j sont deux* grappes interagissantes *ssi*

$$\mathcal{I}_{\Delta_{\gamma_i}}(\gamma_j) \cap \mathcal{I}_{\Delta_{\gamma_j}}(\gamma_i) \neq \emptyset.$$

□

De cette définition, on déduit que γ_i et γ_j sont interagissantes ssi leurs diagnostics épurés sont tels que $\mathcal{I}_{\Delta'_{\gamma_i}}(\gamma_j) = \mathcal{I}_{\Delta'_{\gamma_j}}(\gamma_i) = \mathcal{I}_{\Delta_{\gamma_i}}(\gamma_j) \cap \mathcal{I}_{\Delta_{\gamma_j}}(\gamma_i) \neq \emptyset$. En effet, $\mathcal{I}_{\Delta'_{\gamma_i}}(\gamma_j)$ contient tous les événements de $\mathcal{I}_{\Delta_{\gamma_i}}(\gamma_j)$ qui peuvent être échangés entre γ_i et γ_j selon Δ_{γ_i} et Δ_{γ_j}, ce qui est ni plus ni moins que $\mathcal{I}_{\Delta'_{\gamma_j}}(\gamma_i)$.

Définition 4.15 (Grappes k-interagissantes) *γ_i et γ_j sont deux* grappes k-interagissantes *ssi elles sont interagissantes et*

$$|\mathcal{I}_{\Delta'_{\gamma_i}}(\gamma_j)| = |\mathcal{I}_{\Delta'_{\gamma_j}}(\gamma_i)| = |\mathcal{I}_{\Delta_{\gamma_i}}(\gamma_j) \cap \mathcal{I}_{\Delta_{\gamma_j}}(\gamma_i)| = k.$$

□

Les fusions à privilégier sont celles entre des diagnostics correspondant à des grappes interagissantes. En effet, si les grappes sont interagissantes, cela signifie que des hypothèses de diagnostic de chaque grappe supposent l'échange d'événements communs aux deux grappes. C'est le travail de la fusion de valider ou d'invalider ces hypothèses. La deuxième conséquence est que la fusion de diagnostics correspondant à des grappes γ_1 γ_2 non-interagissantes ne sert à rien. Aucune hypothèse ne sera (in)validée, le résultat de la fusion se résumera au calcul des différentes traces de diagnostics de $\|\gamma_1 \cup \gamma_2\|$, traces qui sont implicitement décrites dans les diagnostics de γ_1 et γ_2.

Définition 4.16 (Diagnostic indépendant) *Un diagnostic Δ_γ est indépendant ssi $\mathcal{I}_{\Delta_\gamma}(\gamma_i) = \emptyset$ pour tout γ_i disjoint de γ.* □

Un diagnostic indépendant est un diagnostic fondé sur une grappe de composants élémentaires γ pour laquelle aucun échange avec des composants extérieurs à γ n'est diagnostiqué (le diagnostic global Δ_Γ est indépendant). Dans ce cas, le diagnostic recense l'ensemble des comportements de γ compatibles avec les observations du système. Tout chemin de transitions d'un diagnostic indépendant participe nécessairement à un chemin de transitions du diagnostic global. Il est donc inutile de fusionner un tel diagnostic avec un autre [8].

Principes et algorithme

L'algorithme 9 résume toute la stratégie de fusion employée en vue d'obtenir le diagnostic global. Cet algorithme de coordination a besoin dans un premier temps des ensembles $\mathcal{T}_i$ et $\mathcal{R}_i$ qui représentent l'ensemble des couples $(e, \mathcal{E}_{\Delta_{\gamma_i}}(e))$ où e est un événement émis ou reçu par $\|\gamma_i\|$ selon Δ_{γ_i} (voir page 117). Il calcule ensuite les événements impossibles (ligne 4) à l'aide de l'algorithme 8 (présenté page 118) en vue de produire des diagnostics locaux épurés (lignes 6-8).

La deuxième phase de l'algorithme consiste à établir les grappes interagissantes. Pour la grappe γ_i, on initialise l'ensemble *k-interactions*(γ_i) (ligne 11). Chaque élément de *k-interactions*(γ_i) est un couple (γ_j, k) informant que γ_i et γ_j sont k-interagissantes.

La troisième phase planifie et applique les fusions de diagnostics. L'ensemble $\mathcal{D}$ contient à tout instant un ensemble de diagnostics ; au départ, il est initialisé avec tous les diagnostics locaux épurés (ligne 14). L'ensemble $\mathcal{P}$ contient l'information sur les interactions entre les différents diagnostics de $\mathcal{D}$. Tant que cet ensemble $\mathcal{P}$ contient des informations sur les interactions, cela implique qu'il existe dans $\mathcal{D}$ au moins deux diagnostics corrrespondant à des grappes interagissantes. Il est donc nécesssaire d'appliquer des fusions. Pour les appliquer, on partitionne $\mathcal{D}$ (ligne 17). La fonction *choisir_partition_de_grappes_interagissantes* établit une partition suivant les critères suivants :

- tout élément de la partition contient au plus deux diagnostics ;
- tout élément de la partition de cardinal supérieur à 1 contient deux diagnostics associés à des grappes k-interagissantes avec *k* aussi grand que possible.

Si un élément de partition contient deux diagnostics, alors ces deux diagnostics correspondent à des grappes interagissantes. Si l'élément de partition ne contient qu'un seul diagnostic, cela signifie que la grappe correspondante n'est interagissante avec aucune grappe (le diagnostic est donc *indépendant*) ou alors les grappes avec lesquelles elle interagit ont leur diagnostic dans un autre élément de partition [9].

Une fois la partition de diagnostic choisie, on fusionne les diagnostics associés à chaque élément de la partition (ligne 18). Nous obtenons un nouvel ensemble de diagnostics (un diagnostic pour chaque élément de partition). Nous mettons à jour les relations d'interactions en

[8] La fusion d'un diagnostic indépendant produirait uniquement un ordonnancement des transitions de ce diagnostic qui soit compatible avec l'ordre partiel global des observations.

[9] Cet élément contient nécessairement deux diagnostics

fonction du nouvel ensemble de diagnostics, et donc du nouvel ensemble de grappes qui leur correspond (ligne 19). Cette mise à jour est établie de la manière suivante :

- soit $\gamma_{i_1} \cup \cdots \cup \gamma_{i_k}$ une grappe de composants associée à un diagnostic de $\mathcal{D}$;
- on crée *k-interactions*$(\gamma_{i_1} \cup \cdots \cup \gamma_{i_k}) = \{(\gamma_{j_1} \cup \cdots \cup \gamma_{j_l}, k), k > 0\}$ où $\gamma_{j_1} \cup \cdots \cup \gamma_{j_l}$ est une autre grappe correspondant à un diagnostic de $\mathcal{D}$ et où

$$k = \sum_{(\gamma_i,\gamma_j)|i\in\{i_1,\ldots,i_k\}j\in\{j_1,\ldots,j_l\}} |\mathcal{I}_{\Delta_{\gamma_i}}(\gamma_j)|.$$

Ensuite, il suffit de réitérer en choisissant une nouvelle partition de diagnostics en fonction des nouvelles interactions. Une fois que l'ensemble $\mathcal{P}$ ne contient plus d'interactions, cela signifie que $\mathcal{D}$ contient un ensemble de diagnostics indépendants représentant implicitement le diagnostic global du système.

Algorithme 9 Calcul du diagnostic global

1: **Entrée :** $\{\Delta_{\gamma_i}, i \in \{1,\ldots,m\}\}$
2: **Entrée :** $\mathcal{T}_1,\ldots,\mathcal{T}_m,\mathcal{R}_1,\ldots,\mathcal{R}_m$
3: {**1– Élimination des hypothèses locales impossibles.**}
4: *événements_impossibles* $\leftarrow$ *détecter_événements_impossibles*$(\mathcal{T}_1,\ldots,\mathcal{T}_m,\mathcal{R}_1,\ldots,\mathcal{R}_m)$
5: {*événements_impossibles* : événements produisant des hypothèses locales impossibles (voir algorithme 8).}
6: **pour tout** $i \in \{1,\ldots,m\}$ **faire**
7: $\quad\Delta'_{\gamma_i} \leftarrow$ *éliminer_hypothèses_impossibles*$(\Delta_{\gamma_i}$, *événements_impossibles* $\cap\, \Sigma^{\gamma_i}_{int})$
8: **fin pour**
9: {**2– Recherche des grappes interagissantes**}
10: **pour tout** $i \in \{1,\ldots,m\}$ **faire**
11: $\quad$*k-interactions*$(\gamma_i) \leftarrow \{(\gamma_j, |\mathcal{I}_{\Delta'_{\gamma_i}}(\gamma_j)|) \mid \mathcal{I}_{\Delta'_{\gamma_i}}(\gamma_j) \neq \emptyset\}$
12: **fin pour**
13: {**3– Planification et application des fusions**}
14: $\mathcal{D} \leftarrow \{\Delta'_{\gamma_i}, i \in \{1,\ldots,m\}\}$
15: $\mathcal{P} \leftarrow \{$*k-interactions*$(\gamma_i), i \in \{1,\ldots,m\}\}$
16: **tant que** $\mathcal{P} \neq \emptyset$ **faire**
17: $\quad\pi_{\mathcal{D}} \leftarrow$ *choisir_partition_de_grappes_interagissantes*$(\mathcal{D},\mathcal{P})$;
18: $\quad\mathcal{D} \leftarrow \bigcup_{\{\Delta_a,\Delta_b\}\in\pi_{\mathcal{D}}}\{\Delta_a \odot \Delta_b\} \cup \bigcup_{\{\Delta_a\}\in\pi_{\mathcal{D}}}\{\Delta_a\}$
19: $\quad\mathcal{P} \leftarrow$ *mettre_à_jour_les_interactions*$(\mathcal{P},\mathcal{D})$
20: **fin tant que**
21: **Sortie :** $\mathcal{D}$

4.5.4 Exemple d'application de la stratégie sur Toynet

Nous présentons ici un exemple simple de l'application de la stratégie sur Toynet. Dans cet exemple, la décentralisation choisie est :

- $\gamma_1 = \{Cnx12, CM1cnx, CM1ctl, SC1\}$;
- $\gamma_2 = \{Cnx23, CM2cnx, CM2ctl, SC2\}$;
- $\gamma_3 = \{Cnx31, CM3cnx, CM3ctl, SC3\}$.

Nous considérons que les canaux de communications sont au nombre de trois, que le superviseur dispose d'un unique capteur datant les alarmes reçues. L'ensemble des observations reçues est le suivant :

- commutateur 1 : *SC1op* $\preceq$ *CM1cx12* $\preceq$ *CM1cx12* ;
- commutateur 2 : *CM1cx12* $\preceq$ *CM2blc* $\preceq$ *CM2op* ;
- commutateur 3 : *SC3op* $\preceq$ *SC3op*.

On considère également qu'il n'y a aucune relation d'ordre entre deux observations de deux canaux différents.

Interactions entre les diagnostics locaux

Une fois les diagnostics locaux calculés (le diagnostic local de γ_1 est identique à celui de la figure 4.2), les interactions sont établies.

- Δ_{γ_1} prétend qu'il peut y avoir des échanges d'événements *CM2attenteCnx12* et *CM2finattenteCnx12* entre γ_1 et γ_2 ;
- Δ_{γ_2} prétend qu'il peut y avoir des échanges d'événements *CM2attenteCnx12*, *CM2finattenteCnx12* entre γ_1 et γ_2 et des échanges d'événements *CM3attenteCnx23*, *CM3finattenteCnx23* entre γ_2 et γ_3 ;
- Δ_{γ_3} prétend qu'il n'a aucun échange d'événements avec les autres groupes de composants, le diagnostic local est indépendant.

La phase d'élimination des hypothèses impossibles consiste donc à éliminer de Δ_{γ_2} les hypothèses de diagnostics supposant l'échange de *CM3attenteCnx23* et de *CM3finattenteCnx23* étant donné que Δ_{γ_3} ne permet pas un tel échange. En fait, selon Δ_{γ_3}, il n'y a pas eu de rupture de connexion entre le commutateur 2 et le commutateur 3. Cette hypothèse de rupture de connexion était possible selon le diagnostic du commutateur 2 car il peut exister des alarmes masquées qui expliquent cette hypothèse (l'alarme *CM2blc* annonce que le commutateur est bloqué, d'où masquage d'alarmes). Cette phase d'élimination a permis de supprimer la moitié des hypothèses de diagnostic de Δ_{γ_2}.

Fusion des diagnostics locaux

Le diagnostic Δ_{γ_3} étant indépendant, il n'est pas fusionné, il fait partie du résultat final. La stratégie impose la fusion de Δ_{γ_1} et de Δ_{γ_2} afin de vérifier que les échanges de *CM2attenteCnx12* et de *CM2finattenteCnx12* sont possibles. Le diagnostic global est donc constitué par l'ensemble $\{\Delta_{\gamma_1} \odot \Delta_{\gamma_2}, \Delta_{\gamma_3}\}$. On peut remarquer que si l'élimination des hypothèses impossibles n'avait pas été effectuée auparavant la fusion de Δ_{γ_2} et de Δ_{γ_3} aurait été nécessaire afin d'invalider les hypothèses de diagnostic de Δ_{γ_2} sur les échanges de *CM3attenteCnx23* et de *CM3finattenteCnx23*.

4.5.5 Résumé

La stratégie de fusion suit le principe du « moindre effort » ; elle peut se résumer en deux étapes. La première consiste à faire un traitement préalable sur les diagnostics locaux. L'idée consiste à extraire des diagnostics locaux une information sur les interactions possibles entre les différents diagnostics locaux. En confrontant ces interactions, on est en mesure de déduire avant toute fusion que certaines hypothèses de diagnostics locales sont invalides. Ce prétraitement est intéressant car il diminue le nombre d'hypothèses locales à valider par la fusion des diagnostics. Le deuxième point est dans l'application de la fusion elle-même. La fusion sert à valider les interactions que les diagnostics locaux proposent. Aussi, les fusions à privilégier sont celles pour lesquelles les diagnostics ont de bonne chance d'avoir de nombreuses interactions à (in)valider. Les fusions de diagnostics qui n'interagissent pas ne sont pas intéressants car elles n'affirment ou n'infirment aucune hypothèse de diagnostic.

4.6 Conclusion

Ce chapitre est une présentation des choix algorithmiques pour la mise en œuvre de l'approche décentralisée du diagnostic. Dans un premier temps, une représentation du diagnostic a été mise en place afin qu'elle soit la plus compacte possible. Cette représentation est à base de transducteurs, ce qui permet d'avoir une représentation finie du diagnostic. Chaque chemin de transitions de ce transducteur représente une *trace* des événements qui ont pu expliquer les observations. Les traces sont un moyen efficace de représenter l'occurrence d'événements qui peuvent se produire en concurrence et qui sont du point de vue du diagnostic, des *événements indépendants*.

Le calcul du diagnostic local peut s'établir de plusieurs manières. Néanmoins, toujours par souci d'efficacité, l'utilisation d'une structure diagnostiqueur est plus intéressante car elle permet la compilation d'un certain nombre d'informations utiles à l'établissement du diagnostic local. Au niveau du calcul du diagnostic global, la mise en place d'une stratégie de fusion est nécessaire. Cette stratégie de fusion est fondée sur les diagnostics locaux et leurs interactions respectives afin d'éviter des calculs qui peuvent parfois être inutiles.

CHAPITRE 5

Incrémentalité

5.1 Introduction

Dans les deux chapitres précédents, nous avions toujours considéré que l'ensemble des observations était connu *a priori*. Cette hypothèse est bien évidemment irréaliste dès lors que l'on cherche à diagnostiquer en ligne des systèmes dynamiques qui fonctionnent en permanence (tel est le cas des réseaux de télécommunications). L'objectif de ce chapitre est de considérer que l'on ne dispose pas à l'instant du diagnostic de toutes les observations du système mais seulement d'une partie. Afin d'assurer un diagnostic le plus fréquemment possible, il faut donc mettre en œuvre une méthode qui fournit un diagnostic, en fonction des observations déjà reçues, et une méthode pour adapter ce diagnostic dès lors que de nouvelles observations apparaissent.

5.2 Diagnostic incrémental : objectifs

5.2.1 Principe

Dans les approches de diagnostic en ligne, l'objectif est de suivre le comportement observable du système et d'informer sur le diagnostic du système. Si nous considérons un instant t pour lequel nous avons établi un diagnostic qui se fonde sur les observations déjà reçues, il devient intéressant de prendre en compte ces résultats afin d'*étendre*, d'*adapter* ce diagnostic si de nouvelles observations se présentent dans le futur et ainsi d'éviter de reconstruire un nouveau diagnostic entièrement. C'est cette problématique que nous nommons *diagnostic incrémental* [Pencolé et al. 01b]. Le diagnostic incrémental s'appuie sur la notion de fenêtre temporelle et de point d'arrêt.

Définition 5.1 (point d'arrêt) *Un* point d'arrêt *est une date issue de l'horloge du superviseur.* □

Définition 5.2 (fenêtre temporelle) *Une* fenêtre temporelle *est l'intervalle de temps entre deux points d'arrêts successifs.* □

Les observations sont considérées comme faisant partie de fenêtres temporelles successives. Ayant déjà établi un diagnostic pour un ensemble de fenêtres temporelles successives, le problème du diagnostic incrémental est d'adapter ce diagnostic afin de prendre en compte les observations de la nouvelle fenêtre temporelle.

Notations

Nous présentons ici un ensemble de notations qui nous serviront tout au long de ce chapitre.
- L'ensemble $\mathcal{O}_{j-1}$ représente toutes les observations qui ont été reçues du début jusqu'au point d'arrêt $j, j \geq 1$.
- Δ_j est le diagnostic du système expliquant $\mathcal{O}_j$.
- $\mathcal{F}_j$ représente la fenêtre temporelle débutant au point d'arrêt j et $\mathcal{O}_{\mathcal{F}_j}$ est l'ensemble des observations reçues durant la fenêtre temporelle $\mathcal{F}_j$.
- $\Delta_{\mathcal{F}_j}$ est le diagnostic de la fenêtre temporelle $\mathcal{F}_j$.
- $P_{\gamma_i}(\mathcal{O})$ est l'ensemble induit de $\mathcal{O}$ contenant toutes les observations issues des composants de γ_i. Par extension, $P_{\delta_i}(\mathcal{O})$ est l'ensemble induit de $\mathcal{O}$ contenant toutes les observations issues des composants diagnostiqués par le diagnostic δ_i.

5.2.2 Problématique

Le problème du diagnostic incrémental peut se poser ainsi. On considère les observations du système. Ces observations sont reçues en séquence et sont munies d'une relation d'ordre partiel liée à leur émission qui est établie à partir des propriétés intrinsèques du système et de son observabilité (voir section 3.3.3 page 65). La séquence d'observations est découpée en sous-séquences, chaque sous-séquence coïncide avec une fenêtre temporelle. Le problème de ce découpage est que l'on ne garantit pas qu'à la fin d'une fenêtre temporelle, l'ensemble des observations émises ont effectivement été reçues. En effet, il peut éventuellement exister des observations qui ont été émises mais qui sont encore en transit sur les canaux de communications.

Exemple La figure 5.1 présente le problème dans le cadre de Toynet. Ce réseau dispose de trois canaux de communications (un par commutateur). Si l'on considère que ces canaux imposent un délai de propagation, alors il peut se présenter une situation identique à celle présentée sur la figure 5.1. Au point d'arrêt t, il y a une observation en transit sur un canal de communication. Les observations reçues indiquent que le commutateur 1 et sa station de contrôle sont opérationnels et que le commutateur 2 détecte un problème de connexion entre les commutateurs 2 et 3. Si l'on tente de fusionner les diagnostics locaux au point d'arrêt t, le résultat de cette fusion sera un diagnostic nul. En effet, l'ensemble des observations n'est pas complet. D'après le modèle, pour pouvoir conclure, le superviseur doit « attendre » une observation du commutateur 3 :
- soit *CM3cx23*, dans ce cas le commutateur 3 n'est pas bloqué, il a lui-même détecté le problème de connexion ;
- soit *CM3blc*, dans ce cas il s'est bloqué et il n'a pas pu détecter la rupture de connexion.

Cet exemple est un cas atypique où le diagnostic résultat de la fusion est nul. En règle générale, la fusion détermine un diagnostic mais un certain nombre d'hypothèses comme celle présentée ci-dessus sont oubliées, ce qui n'est pas satisfaisant.

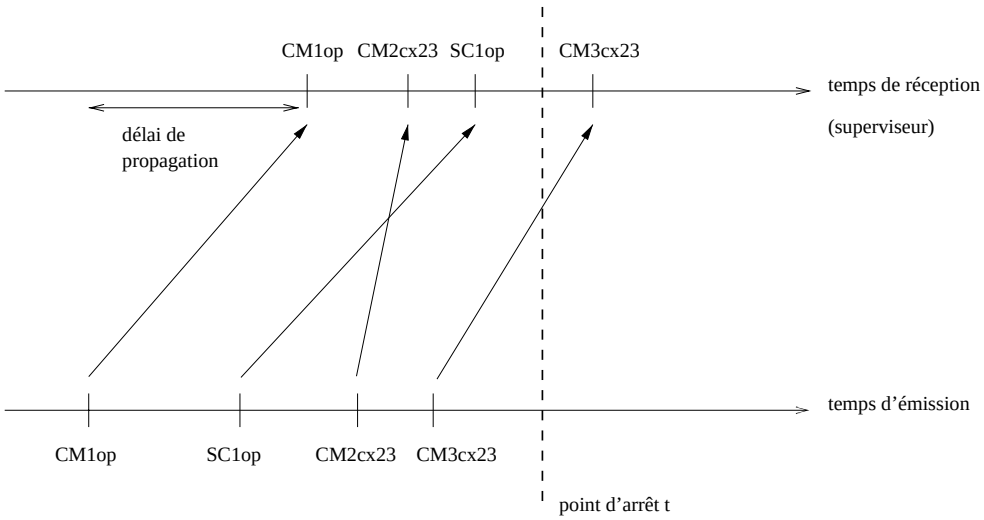

FIG. 5.1 – Point d'arrêt

Le choix d'une fenêtre temporelle est donc très important. De celui-ci dépend la nature de l'adaptation du diagnostic global dès lors que l'on traite une nouvelle fenêtre temporelle. Des discussions sur ce sujet se trouvent également dans [Aghasaryan 98] et dans [Debouk et al. 00b].

Dans la suite de ce chapitre, l'adaptation du diagnostic est présentée dans deux cas différents.

1. Le premier cas est fondé sur le fait qu'on peut garantir que chaque fenêtre temporelle satisfait la propriété de complétude des observations.
2. Le deuxième cas est général : les fenêtres temporelles ne garantissent pas la propriété de complétude.

5.3 Algorithme incrémental dans des fenêtres sûres

La première solution consiste à choisir des fenêtres temporelles qui satisfont la propriété de complétude des observations au point d'arrêt sélectionné.

5.3.1 Notion de fenêtres sûres

Définition 5.3 *Une fenêtre temporelle $\mathcal{F}_j$ est* sûre *par rapport à un ensemble d'observations $\mathcal{O}_{j-1}$ ssi $\forall o_2 \in \mathcal{O}_{\mathcal{F}_j}, \forall o_1 \in \mathcal{O}_{j-1}$, o_1 a été reçue avant l'émission de o_2. Soit t la date de*

réception de la dernière observation de $\mathcal{O}_{j-1}$, on dit que t est un point d'arrêt sûr. □

Un point d'arrêt sûr est intéressant car il permet de s'assurer que l'ensemble des observations émises avant ce point a été effectivement reçu. Du point de vue de l'ordre partiel des observations, cela implique que pour toute observation o_1 reçue avant un point d'arrêt sûr, pour toute observation o_2 émise et reçue après ce point d'arrêt, on a $o_1 \preceq o_2$ et donc :

$$\mathcal{O}_{j-1} \sqsubseteq \mathcal{O}_j.$$

Exemple La figure 5.2 présente un point d'arrêt sûr. Si l'on suppose que *CM3blc*, *CM2op*, *CM1cx12* et *CM2cx23* ont été émises après la réception de *CM2blc*, la date de réception t de *CM2blc* est un point sûr.

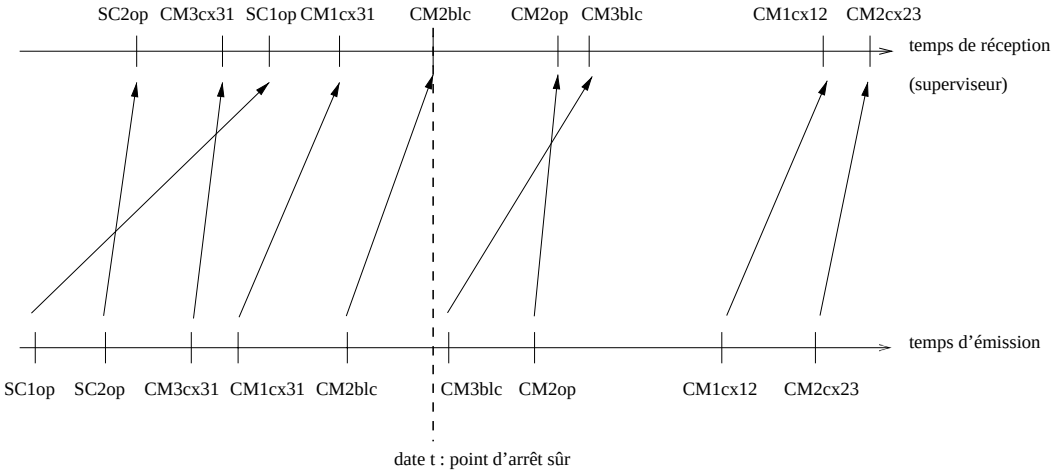

FIG. 5.2 – Point d'arrêt sûr : la date t.

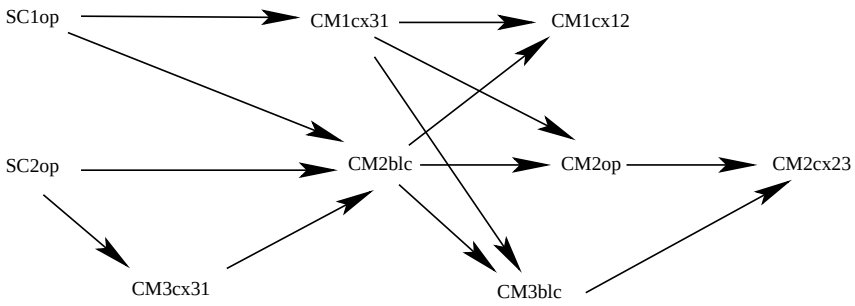

FIG. 5.3 – Ordre partiel d'observations associé à la figure 5.2.

La figure 5.3 présente un ordre partiel sur l'émission des observations dont une séquence possible est celle présentée sur la figure 5.2 dans le cas où la date t est un point d'arrêt sûr. Toute observation reçue avant t est nécessairement émise avant *CM1cx12*, *CM2op* et *CM3blc*. Par fermeture transitive de la relation d'ordre partiel, on peut garantir que toute observation reçue avant la date t est en relation de précédence avec toute observation reçue après la date t.

5.3.2 Calcul de $\Delta_{\mathcal{F}_j}$

$\Delta_{\mathcal{F}_j}$ est établi par l'algorithme 10. Le diagnostic $\Delta_{\mathcal{F}_j}$ est établi à partir du diagnostic de la fenêtre temporelle précedente : $\Delta_{\mathcal{F}_{j-1}}$. Ce diagnostic est constitué d'un ensemble de diagnostics indépendants δ_i issus de l'application de la stratégie de fusion (voir algorithme 9 page 121). L'idée consiste à extraire de ces δ_i l'ensemble des états courants de tous les γ_j où γ_j est un élément de la décentralisation du modèle (voir section 3.4.2.2 page 75). Calculer le diagnostic $\Delta_{\mathcal{F}_j}$ de la fenêtre temporelle courante consiste alors à appliquer la stratégie de fusion sur les diagnostics locaux associés aux observations de la fenêtre courante avec l'hypothèse pour chaque γ_i que les états initiaux au début de la fenêtre $\mathcal{F}_j$ sont les états de *initiaux*(γ_i). Le résultat de cette fusion est un nouvel ensemble de diagnostics indépendants δ'_i.

Algorithme 10 Calcul de $\Delta_{\mathcal{F}_j}$

Entrée 1 : $\Delta_{\mathcal{F}_{j-1}} = \{\delta_1, \ldots, \delta_l\}$

Entrée 2 : $\mathcal{O}_{\mathcal{F}_j}$

pour tout $i \in \{1, \ldots, l\}$ **faire**

  pour tout $\gamma_k = (I_{\gamma_k}, O_{\gamma_k}, Q_{\gamma_k}, E_{\gamma_k})$ diagnostiqué par $\delta_i = (I_{\delta_i}, O_{\delta_i}, Q_{\delta_i}, E_{\delta_i})$ **faire**

    initiaux$(\gamma_k) \leftarrow \{q_k | q_k \in Q_{\gamma_k} \wedge ((\ldots, q_k, \ldots), P_{\delta_i}(\mathcal{O}_{\mathcal{F}_{j-1}})) \in Q_{\delta_i}\}$

  fin pour

fin pour

$\{\delta'_1, \ldots, \delta'_p\} \leftarrow$ *appliquer_fusion*$(\Delta_{\gamma_1}($*initiaux*$(\gamma_1), P_{\gamma_1}(\mathcal{O}_{\mathcal{F}_j})), \ldots,$
$\Delta_{\gamma_m}($*initiaux*$(\gamma_m), P_{\gamma_m}(\mathcal{O}_{\mathcal{F}_j})))$

$\{\Delta_{\gamma_k}(\text{initiaux}(\gamma_k), P_{\gamma_k}(\mathcal{O}_{\mathcal{F}_j}))$ diagnostic local à γ_k expliquant $P_{\gamma_k}(\mathcal{O}_{\mathcal{F}_j})$ à partir de initiaux(γ_k) (voir notation page 97).$\}$

Sortie : $\Delta_{\mathcal{F}_j} = \{\delta'_1, \ldots, \delta'_p\}$

Chaque diagnostic indépendant représente l'ensemble des comportements possibles d'un groupe de composants élémentaires. Dans ce nouvel ensemble de diagnostics indépendants, le groupe de composants élémentaires associé à l'un des diagnostics δ'_j peut n'avoir aucun rapport avec un groupe de composants élémentaires associé à un diagnostic indépendant δ_k de la fenêtre temporelle précédente. Il se peut en effet que lors de la fenêtre $\mathcal{F}_j$, les grappes interagissantes utilisées au cours de l'application de la stratégie de fusion ne soient pas les mêmes. Elles dépendent en effet des interactions locales qui ont été diagnostiquées au cours de cette fenêtre uniquement.

5.3.3 Raffinement du diagnostic

Une fois le diagnostic $\Delta_{\mathcal{F}_j}$ établi, on est en mesure de connaître les états courants du système à la fin de la fenêtre $\mathcal{F}_j$. Le problème est que ce nouveau diagnostic $\Delta_{\mathcal{F}_j}$ a pu invalider certaines hypothèses de Δ_{j-1} : en effet toute hypothèse de Δ_{j-1} pour laquelle il n'existe pas dans $\Delta_{\mathcal{F}_j}$ une continuation qui explique les observations $\mathcal{O}_{\mathcal{F}_j}$ doit être éliminée. Ainsi, le diagnostic Δ_j est obtenu en raffinant Δ_{j-1} et en le concaténant avec $\Delta_{\mathcal{F}_j}$.

Algorithme 11 Algorithme de raffinement : $\Delta_j = \Delta_{j-1} \oplus \Delta_{\mathcal{F}_j}$

Entrée 1 : Diagnostic des fenêtres passées Δ_{j-1}

Entrée 2 : Diagnostic de la fenêtre courante $\Delta_{\mathcal{F}_j} = \{\delta'_1, \ldots, \delta'_p\}$

{**Concaténation des diagnostics**}

$\Delta_{tmp} \leftarrow$ *concaténer_diagnostics*$(\Delta_{j-1}, \bigodot_{i=1}^{p} \delta'_i)$

{**Élimination des hypothèses qui n'expliquent pas toutes les observations** $\mathcal{O}_j$}

pour tout $X = (q, \mathcal{O}_{j-1}) \in \Delta_{tmp}$ **faire**

si $\nexists X \xrightarrow{t} X' \in \Delta_{tmp} \wedge |OBS(X \xrightarrow{t} X')| > 0$ **alors**

$\{X$ est un état final de Δ_{j-1} qui doit être éliminé, ainsi que les chemins de transitions y accédant$\}$

$\Delta_{tmp} \leftarrow$ *éliminer_chemins*(Δ_{tmp}, X)

fin si

fin pour

Sortie : $\Delta_j \leftarrow \Delta_{tmp}$

L'algorithme 11 présente le calcul de Δ_j en fonction de Δ_{j-1} et de $\Delta_{\mathcal{F}_j} = \{\delta'_1, \ldots, \delta'_p\}$. On peut remarquer que cet algorithme est sujet à un problème d'efficacité. En effet, la concaténation du diagnostic Δ_{j-1} avec celui de la fenêtre courante est une concaténation de transducteurs [Aho et Ullman 72] et nécessite de fusionner les diagnostics indépendants de $\Delta_{\mathcal{F}_j}$. Cette fusion est nécessaire car les indépendances détectées dans $\Delta_{\mathcal{F}_j}$ peuvent ne pas correspondre à celles détectées dans les fenêtres précédentes (la concaténation de diagnostics indépendants qui ne diagnostiquent pas les mêmes groupes de composants n'a pas de sens).

Le principe de la concaténation est le suivant. Tout état et toute transition de Δ_{j-1} sont inclus dans Δ_{tmp}. Ensuite, on considère un état final $X = ((q_1, \ldots, q_n), OBS(\mathcal{C}) \diamond \mathcal{O}_{j-1})$ de Δ_{j-1} où $\mathcal{C}$ est un chemin de transitions de $\|\Gamma\|$ menant à $(q_1, \ldots, q_n)$ et expliquant $\mathcal{O}_{j-1}$. On considère l'état $((q_1, \ldots, q_n), \emptyset)$ de $\Delta_{\mathcal{F}_j}$ qui lui est associé (s'il existe). Pour tout état $X' = ((q'_1, \ldots, q'_n), OBS(\mathcal{C}') \diamond \mathcal{O}')$ de $\Delta_{\mathcal{F}_j}$ ($\mathcal{O}' \sqsubseteq \mathcal{O}_{\mathcal{F}_j}$ et $\mathcal{C}'$ chemin de transitions de $(q_1, \ldots, q_n)$ à $(q'_1, \ldots, q'_n)$ dans $\|\Gamma\|$ expliquant $\mathcal{O}'$), on construit dans Δ_{tmp} l'état $((q'_1, \ldots, q'_n), OBS(\mathcal{C}.\mathcal{C}') \diamond (\mathcal{O}_{j-1}.\mathcal{O}'))$ où $\mathcal{O}_{j-1}.\mathcal{O}'$ est l'ensemble partiellement ordonné défini par

$$\forall o \in \mathcal{O}_{j-1}.\mathcal{O}', o \in \mathcal{O}_{j-1} \vee o \in \mathcal{O}'$$

muni de la relation d'ordre $\preceq$ définie par

$$o_1 \preceq o_2 \equiv (o_1, o_2 \in \mathcal{O}_{j-1} \wedge o_1 \preceq_{\mathcal{O}_{j-1}} o_2) \vee (o_1, o_2 \in \mathcal{O}' \wedge o_1 \preceq_{\mathcal{O}'} o_2) \vee (o_1 \in \mathcal{O}_{j-1} \wedge o_2 \in \mathcal{O}').$$

L'ordre partiel $\mathcal{O}_{j-1}.\mathcal{O}'$ correspond effectivement à l'ordre partiel des observations reçues dans l'état $(q'_1, \ldots, q'_n)$ puisque la fenêtre $\mathcal{F}_j$ est considérée comme sûre.

Calcul du raffinement en pratique

Le raffinement de Δ_j est une opération coûteuse car elle nécessite la fusion de diagnostics indépendants. Pour le calcul de $\Delta_{\mathcal{F}_j}$, ce raffinement n'est pas nécessaire car il ne nécessite que la connaissance du diagnostic de la fenêtre précédente. Par conséquent, en pratique, on pourra ne pas effectuer le raffinement lors du suivi en ligne du système supervisé.

Le raffinement est néanmoins nécessaire car il permet d'éliminer des hypothèses de diagnostics passées invalidées par de nouvelles observations. On pourra alors effectuer le raffinement en vue de faire des études globales sur le comportement du système (taux de pannes...) dans un cadre hors-ligne.

5.4 Algorithme incrémental dans le cas général

5.4.1 Introduction

Il n'est pas toujours possible de détecter des points d'arrêts sûrs. Il se peut en effet qu'il n'en existe pas. Cette section considère donc le cas général où il n'est pas garanti que les fenêtres sélectionnées sont sûres. Dans ce cadre général, la difficulté du diagnostic incrémental est de considérer deux types d'observations :

1. les observations reçues pendant la fenêtre temporelle courante ;
2. les événements émis par le système pendant la fenêtre temporelle (et même avant) et qui n'ont pas encore été reçus. Ces événements sont encore dans les canaux de communications.

Autrement dit, dans le cas général, si on établit le diagnostic d'une fenêtre temporelle avec l'algorithme 10, on va oublier des hypothèses de diagnostics car les états courants d'un tel diagnostic considèrent que toutes les observations ont été effectivement reçues. Afin d'avoir une approche efficace pour mettre à jour le diagnostic du système, on veut toujours pouvoir établir le diagnostic de la fenêtre courante à partir des états courants du diagnostic de la fenêtre précédente. Il faut donc que le diagnostic d'une fenêtre temporelle prenne en compte les observations potentiellement émises mais non reçues : ce type de diagnostic est appelé *diagnostic étendu*. Dans la suite, l'hypothèse suivante est nécessaire.

Hypothèse 5.1 *On considère ici que les canaux de communications entre le système supervisé et le superviseur sont modélisables par des files bornées.* □

5.4.2 Diagnostic local étendu

Le diagnostic local *étendu* $\Delta^{etd}_{\gamma_i}(\mathcal{F}_j)$ sur la fenêtre temporelle $\mathcal{F}_j$ dépend des états de $\|\gamma_i\|$ décrits dans les états du diagnostic étendu $\Delta^{etd}_{\mathcal{F}_{j-1}}$ (diagnostic étendu expliquant les observations $\mathcal{O}_{\mathcal{F}_{j-1}}$ plus éventuellement celles que l'on a supposé ne pas avoir encore reçues à la fin de la fenêtre $\mathcal{F}_{j-1}$). Un tel état est donc du type $X = ((q_1, \ldots, q_m), OBS(\mathcal{C}) \diamond P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}}))$ où $\mathcal{C}$ est un chemin de transitions de $\|\gamma_i\|$ et $P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}})$ est l'ensemble partiellement ordonné des observations de γ_i induit de $\mathcal{O}^{etd}_{\mathcal{F}_{j-1}}$, l'ensemble des observations étendu diagnostiqué dans la fenêtre $\mathcal{F}_{j-1}$.

Nous présentons l'adaptation du diagnostic local étendu dans un cas simple que nous généraliserons ensuite.

5.4.2.1 Cas où γ_i dispose d'un seul canal de communication

Dans ce cas, on suppose que les observations reçues par le superviseur et venant de γ_i sont telles que $o_1 \preceq o_2$ si o_1 a a été reçue par le superviseur avant o_2. On supposera également que le nombre maximal d'événements en transit sur la file de γ_i est k_i.

En regardant les observations de γ_i reçues lors de la fenêtre temporelle $\mathcal{F}_j$, on peut vérifier si la supposition que l'on a effectuée dans l'état $X = ((q_1, \ldots, q_m), OBS(\mathcal{C}) \diamond P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}}))$ est valide ou non. D'après l'hypothèse 5.1, deux cas se présentent. Si on note par $P_{\gamma_i}(\mathcal{O})$ l'ensemble partiellement ordonné des observations induit de $\mathcal{O}$ contenant toutes les observations émises par des composants de γ_i, les deux cas sont les suivants.

1. $|P_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j})| < k_i$. $P_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j})$ est totalement expliqué par les suppositions faites dans $P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}})$. Il reste éventuellement un ensemble d'observations de $P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}})$ qui n'ont pas encore été reçues, elles sont au nombre de $k_i - |P_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j})|$. Au plus $|P_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j})|$ observations ont été émises durant la fenêtre $\mathcal{F}_j$.
2. $|P_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j})| \geq k_i$. L'ensemble supposé émis dans la fenêtre $\mathcal{F}_{j-1}$ a été observé et d'autres observations ont été émises et reçues durant la fenêtre $\mathcal{F}_j$. Au plus k_i observations ont été émises durant la fenêtre $\mathcal{F}_j$.

Dans le premier cas, X est un état résultant d'hypothèses de diagnostic expliquant déjà les observations $P_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j})$. Il suffit alors de déterminer les hypothèses de diagnostic local issues de l'état X qui expliquent un certain nombre d'événements non reçus qui peuvent avoir été émis durant la fenêtre $\mathcal{F}_j$ et non reçus durant cette même fenêtre.

Dans le second cas, il faut déterminer les hypothèses de diagnostic local qui expliquent les observations de $P_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j})$ qui ne sont pas dans $P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}})$, suivies par d'hypothétiques événements émis durant $\mathcal{F}_j$ et non reçus dans cette même fenêtre.

Par conséquent, dans les deux cas, il faut déterminer les événements qui peuvent éventuellement ne pas être reçus au cours de la fenêtre temporelle $\mathcal{F}_j$.

Définition 5.4 (*non_reçus*$_{\gamma_i}(k)$) *Soit k un entier positif, l'ensemble non_reçus$_{\gamma_i}(k)$ est l'ensemble des ensembles partiellement ordonnés $OBS_{\gamma_i}(\mathcal{C}_i)$ tels que : $\mathcal{C}_i$ est un chemin de transitions de $\|\gamma_i\|$ et $|OBS_{\gamma_i}(\mathcal{C}_i)| \leq k$.* □

Ainsi, à partir de l'état $X = (q_1, \ldots, q_m)$, il faut établir les hypothèses locales de diagnostic de γ_i qui expliquent l'ensemble des comportements observables de $CompObs_{\gamma_i}(q_i, k_i)$ défini comme suit.

– *Premier cas* : chaque comportement de $CompObs_{\gamma_i}(q_i, k_i)$ appartient à :

$$\textit{non\_reçus}_{\gamma_i}(|\mathcal{P}_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j})|).$$

– *Deuxième cas* : chaque comportement obs_i de $CompObs_{\gamma_i}(q_i, k_i)$ est tel que :

$$obs_i = obs_i^1.obs_i^2$$

où $obs_i^1 = \mathcal{P}_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j}) \setminus P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}})$ est l'ensemble des observations de $\mathcal{F}_j$ qui n'ont pas été expliquées durant la fenêtre $\mathcal{F}_{j-1}$ (obs_i^1 est un ordre total du fait que le canal est une file) et $obs_i^2 \in \textit{non\_reçus}_{\gamma_i}(k_i)$ est un comportement observable d'au plus k_i événements observables pouvant avoir été émis mais pas encore reçus [1].

5.4.2.2 Cas où γ_i dispose de plusieurs canaux de communications

Dans le cas général, γ_i peut disposer de plusieurs canaux de communications vers le superviseur. Dans ce cas, il faut considérer dans le diagnostic local étendu les événements qui peuvent avoir été émis sur les différents canaux et non reçus. Dans cette section, on considère que γ_i dispose de l canaux ($l \geq 2$) que l'on nomme $c_1, \ldots, c_l$. La taille maximale du canal c_j est k_{c_j}. On considère toujours que l'on a un état du type $X = ((q_1, \ldots, q_m), OBS(\mathcal{C}) \diamond P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}}))$ du diagnostic de la fenêtre temporelle précédente, $(q_1, \ldots, q_m)$ un état de $\|\gamma_i\|$.

On note par $\mathcal{P}_{\gamma_i,c_j}(\mathcal{O}_{\mathcal{F}_j})$ l'ensemble induit de $\mathcal{O}_{\mathcal{F}_j}$ qui représente les observations reçues du canal c_j pendant la fenêtre $\mathcal{F}_j$. On note également par $\textit{non\_reçus}^{c_j}_{\gamma_i}(k)$ l'ensemble des ensembles partiellement ordonnés d'observations (notés OBS_{c_j}) induits de $OBS_{\gamma_i}(\mathcal{C})$ tels que : $\mathcal{C}$ est un chemin de transitions de $\|\gamma_i\|$ et $|OBS_{c_j}| \leq k$. Tout ensemble de $\textit{non\_reçus}^{c_j}_{\gamma_i}(k)$ est un ordre partiel d'observations de γ_i de cardinalité au plus égale à k qui sont émises par le canal c_j.

L'ensemble des comportements observables possibles $CompObs_{\gamma_i}(q_i, \sum_{j=1}^{l} k_{c_j})$ issus de γ_i est défini de la façon suivante. Chaque ensemble partiellement ordonné $CompObs$ de $CompObs_{\gamma_i}(q_i, \sum_{j=1}^{l} k_{c_j})$ contient l'ensemble partiellement ordonné des observations $\mathcal{P}_{\gamma_i}(\mathcal{O}_{\mathcal{F}_j}) \setminus P_{\gamma_i}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}})$ reçues pendant la fenêtre $\mathcal{F}_j$ et qui n'ont pas été expliquées durant la fenêtre $\mathcal{F}_{j-1}$. Dans cet ensemble $CompObs$, pour chaque canal c_j, on y inclut également un des ensembles suivants.

– Un ensemble d'observations $\textit{non\_reçus}^{c_j}_{\gamma_i}(|\mathcal{P}_{\gamma_i,c_j}(\mathcal{O}_{\mathcal{F}_j})|)$ si $|\mathcal{P}_{\gamma_i,c_j}(\mathcal{O}_{\mathcal{F}_j})| < k_{c_j}$. La relation de précédence temporelle de $CompObs$ est telle que toute observation de $\mathcal{P}_{\gamma_i,c_j}(\mathcal{O}_{\mathcal{F}_j}) \setminus P_{\gamma_i,c_j}(\mathcal{O}^{etd}_{\mathcal{F}_{j-1}})$ précède les observations supposées émises de $\textit{non\_reçus}^{c_j}_{\gamma_i}(|\mathcal{P}_{\gamma_i,c_j}(\mathcal{O}_{\mathcal{F}_j})|)$ (ce qui traduit le fait que c_j est un file).

– Un ensemble d'observations supposées émises $\textit{non\_reçus}^{c_j}_{\gamma_i}(k_{c_j})$ si $|\mathcal{P}_{\gamma_i,c_j}(\mathcal{O}_{\mathcal{F}_j})| \geq k_{c_j}$.

De plus, on considère dans $CompObs$ qu'il n'y a pas de relation d'ordre entre deux observations supposées émises et qui transitent sur deux canaux différents. Ceci permet de garantir que le diagnostic local étendu n'oubliera pas d'hypothèses.

[1] Pour la notation pointée entre deux ensembles partiellement ordonnés, voir page 130.

5.4.2.3 Calcul du diagnostic étendu

L'algorithme 12 présente le calcul du diagnostic étendu $\Delta^{etd}_{\mathcal{F}_j}$ pour une fenêtre temporelle $\mathcal{F}_j$ donnée. Il nécessite la connaissance du diagnostic étendu précédent $\Delta^{etd}_{\mathcal{F}_{j-1}}$ et des observations de la fenêtre $\mathcal{F}_j$. Ce diagnostic est établi en appliquant à l'identique la stratégie de fusion sur les diagnostics locaux étendus.

Chaque diagnostic local étendu $\Delta^{etd}_{\gamma_i}(\mathcal{F}_j)$ est constitué de l'ensemble des hypothèses locales établies à partir de chaque état courant proposé par le diagnostic de la fenêtre précédente $\Delta^{etd}_{\mathcal{F}_{j-1}}$, chaque hypothèse devant expliquer l'ensemble des événements $CompObs_{\gamma_i}(q_i, k_i)$. Tout état du diagnostic local expliquant ces événements est possible à la fin de la fenêtre temporelle $\mathcal{F}_j$.

Algorithme 12 Calcul du diagnostic étendu de $\mathcal{F}_j$: $\Delta^{etd}_{\mathcal{F}_j}$

Entrée 1 : $\mathcal{O}_{j-1}$, $\mathcal{O}_{\mathcal{F}_j}$

Entrée 2 : $\Delta^{etd}_{\mathcal{F}_{j-1}} = \{\delta_1, \ldots, \delta_l\}$

pour tout $i \in \{1, \ldots, l\}$ **faire**

  pour tout $\gamma_k = (I_{\gamma_k}, O_{\gamma_k}, Q_{\gamma_k}, E_{\gamma_k})$ diagnostiqué par $\delta_i = (I_{\delta_i}, O_{\delta_i}, Q_{\delta_i}, E_{\delta_1})$ **faire**

    {Calcul du diagnostic local étendu de γ_k}

    $\Delta_{tmp} \leftarrow \emptyset$

    pour tout $\{q_k | q_k \in Q_{\gamma_k} \wedge ((\ldots, q_k, \ldots), OBS(\mathcal{C}_\delta) \diamond P_{\delta_i}(\mathcal{O}^{etd}_{j-1})) \in Q_{\delta_i}\}$ **faire**

      {q_k est un état possible de γ_k après la fenêtre $\mathcal{F}_{j-1}$ selon $\Delta^{etd}_{\mathcal{F}_{j-1}}$.}

      $\Delta_{tmp} \leftarrow \Delta_{tmp} \cup \Delta_{\gamma_k}(\{q_k\}, CompObs_{\gamma_k}(q_k, k_k))$

      {$\Delta_{\gamma_k}(\{q_k\}, CompObs_{\gamma_k}(q_k, k_k))$: diagnostic local de γ_k expliquant $CompObs_{\gamma_k}(q_k, k_k))$ à partir de l'état q_k (voir notation page 97).}

    fin pour

    $\Delta^{etd}_{\gamma_k}(\mathcal{F}_j) \leftarrow \Delta_{tmp}$

  fin pour

fin pour

$\{\delta'_1, \ldots, \delta'_p\} \leftarrow$ *appliquer_fusion*$(\Delta^{etd}_{\gamma_1}(\mathcal{F}_j), \ldots, \Delta^{etd}_{\gamma_m}(\mathcal{F}_j))$

Sortie : $\Delta^{etd}_{\mathcal{F}_j} = \{\delta'_1, \ldots, \delta'_p\}$

Exemple La figure 5.4 présente un ensemble d'observations de Toynet. Cet ensemble est découpé en 3 fenêtres temporelles. On considère dans cet exemple que Toynet est constitué de 3 canaux de communications (files) de taille 1. Autrement dit, on suppose qu'à tout instant un seul événement observable peut transiter sur un canal. Si l'on cherche à établir un diagnostic global à la fin de la fenêtre $F1$, on se retrouve dans la situation présentée sur la figure 5.1. Le diagnostic étendu à la fin de $F1$ explique les observations reçues dans la fenêtre. Il explique de

plus des observations qui ont pu être émises à la fin de $F1$ et qui ne sont pas encore reçues. Au plus, il explique le cas où il existe une observation transitant dans chaque canal à la fin de $F1$. En particulier, il explique le cas où $CM3ctl$ a émis $CM3cx23$ et le cas où il a émis $CM3blc$. À la fin de $F1$, le diagnostic étendu contient l'ensemble des hypothèse possibles à cette date. À la fin de $F2$, aucune observation n'a été reçue sur le canal de $CM1$, le diagnostic étendu de la fenêtre précédente avait diagnostiqué la possibilité d'une émission sur ce canal, cette supposition est toujours valide. Sur le canal de $CM2$, seules les hypothèses où $CM2blc$ avait été supposé émis à la fin de $F1$ sont vérifiées, on les adapte afin qu'elles expliquent désormais qu'une nouvelle observation a pu être émise après $CM2blc$. Sur le canal de $CM3$, seules les hypothèses supposant l'émission de $CM3cx23$ à la fin de $F1$ sont correctes, on les adapte afin qu'elles expliquent la deuxième occurrence de $CM3cx23$ et une observation émise durant $F2$ mais pas encore reçue.

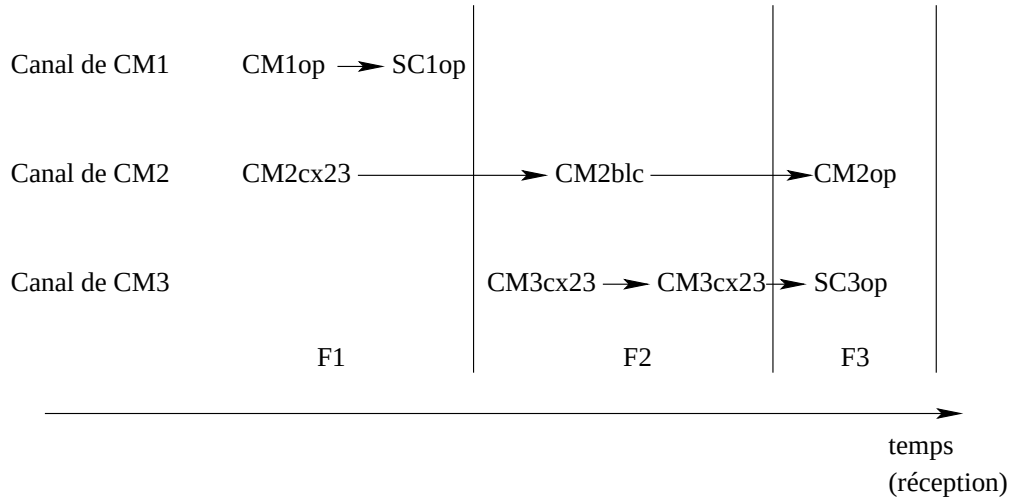

FIG. 5.4 – Ensemble de fenêtres temporelles.

5.4.3 Mise à jour du diagnostic global

Le diagnostic global Δ_j est établi de la même façon que dans le cadre des fenêtres sûres. Il suffit d'appliquer l'algorithme 11, on a donc :

$$\Delta_j^{etd} = \Delta_{j-1}^{etd} \oplus \Delta_{\mathcal{F}_j}^{etd}.$$

Cette opération est néanmoins sujette à une difficulté supplémentaire durant la phase de concaténation, qui est liée aux suppositions faites sur l'émission d'événements observables non reçus. Considérons en effet un état de Δ_{j-1}^{etd}, cet état est du type $X = ((q_1, \ldots, q_n), OBS(\mathcal{C}) \diamond \mathcal{O}_{j-1}^{etd})$. Les observations de la nouvelle fenêtre $\mathcal{F}_j$ sont plus ou moins partiellement expliquées par $OBS(\mathcal{C}) \diamond \mathcal{O}_{j-1}^{etd}$. Par contre, l'ordre partiel que l'on a supposé est moins strict que celui qui a été observé. Autrement dit, l'état $X = ((q_1, \ldots, q_n), OBS(\mathcal{C}) \diamond \mathcal{O}_{j-1}^{etd})$ de Δ_{j-1}^{etd} peut

être l'extrémité d'un comportement expliquant une séquence d'observations dont l'ordonnancement n'est pas compatible avec l'ordre partiel observé. Lors de la phase de concaténation, il faut donc considérer ce problème et éliminer les chemins de transitions de Δ_{j-1}^{etd} qui mènent à q et qui expliquent un ordonnancement des observations incompatible avec les observations de la fenêtre $\mathcal{F}_j$. Il se peut même que plus aucun chemin de transitions n'accède à q : l'état q doit être éliminé définitivement.

Relations entre Δ_j et Δ_j^{etd}

Après l'utilisation de l'algorithme de raffinement sur une fenêtre temporelle $\mathcal{F}_j$, le diagnostic courant Δ_j^{etd} décrit les hypothèses qui expliquent $\mathcal{O}_j$. Certaines de ces hypothèses expliquent de plus des événements qui sont supposés avoir été émis mais qui ne sont pas encore reçus (ils sont dans les canaux de communications). On a donc la relation suivante :

$$\Delta_j \subseteq \Delta_j^{etd}.$$

En définitive, si l'on considère que, après une fenêtre temporelle $\mathcal{F}_{fin}$, il n'y a plus d'observation possible, on considère alors que $\mathcal{O}_{fin}$ est l'ensemble complet des observations. Dans ce cas, on peut calculer le diagnostic étendu en supposant qu'aucun événement ne se situe dans les canaux (paramètres $k = 0$). Par conséquent, les suppositions sur l'envoi d'événements non reçus qui ont été effectuées durant les différentes fenêtres temporelles successives sont éliminées grâce à cette dernière fenêtre, et on a finalement :

$$\Delta_{fin} = \Delta_{fin}^{etd}.$$

5.5 Conclusion

Le diagnostic incrémental est essentiel si l'on veut que le système de diagnostic soit en mesure d'effectuer un suivi du système supervisé et d'établir un diagnostic le plus fréquemment possible. Dans un contexte de diagnostic en ligne, les observations sont considérées comme appartenant à des fenêtres temporelles successives. Deux réponses à ce problème ont été examinées.

La première solution consiste à établir des points d'arrêts sûrs qui déterminent des fenêtres temporelles sûres. Dans cette solution, le calcul du diagnostic du système pour la fenêtre courante est simple à mettre en place, il ne nécessite que la connaissance du diagnostic de la fenêtre précédente. Le calcul de points d'arrêts sûrs n'est possible qu'en utilisant des connaissances sur l'application étudiée, plus particulièrement, les connaissances sur les propriétés des canaux de communications. Par exemple, si le délai maximal d_{max} de propagation des événements dans les canaux de communications est connu, si à partir d'une date t, on ne reçoit rien pendant un délai d_{max}, alors on est assuré que t est un point d'arrêt sûr.

Malheureusement, il n'est pas toujours possible de détecter des points d'arrêts que l'on garantit comme étant sûrs. Dans le cas général, afin de garantir que le diagnostic de la fenêtre courante ne nécessite que la connaissance du diagnostic de la fenêtre précédente, il faut émettre des hypothèses sur l'émission d'événements par le système qui sont observables mais qui n'ont

pas encore été observés. Un tel diagnostic, nommé diagnostic étendu, utilise le même principe de fusion de diagnostics locaux, seul l'ensemble des observations est modifié.

Ces deux solutions peuvent bien évidemment être combinées. Si au bout de m fenêtres temporelles, on se rend compte que la dernière est sûre, il suffit de changer d'algorithme pour établir le diagnostic de la dernière fenêtre.

CHAPITRE *6*

Ddyp : une plate-forme de diagnostic

6.1 Introduction

Tous les travaux présentés dans les chapitres précédents ont abouti à la réalisation d'une plate-forme de diagnostic : *Ddyp* [1]. Cette plate-forme met en œuvre tous les aspects de l'approche décentralisée du diagnostic. Elle a permis en particulier de valider cette approche sur des cas concrets de gestion de réseaux de télécommunications : la gestion d'une partie du réseau Transpac et celle d'un réseau de type SDH.

6.2 Présentation du logiciel

Ddyp est une plate-forme qui met en œuvre tous les outils nécessaires à la mise en place d'un processus de diagnostic décentralisé en-ligne sur un système à événements discrets réparti [Pencolé et al. 01a]. Ces outils peuvent être classés de la façon suivante :

- *modélisation* : cette partie a pour objectif de mettre en œuvre le modèle du système à superviser qui est le point d'entrée unique de la plate-forme ;
- *diagnostic* : cela constitue le cœur du système, il s'agit en effet de l'architecture logicielle mettant en œuvre l'approche décentralisée du diagnostic et proposant à partir de flux d'observations un diagnostic du système supervisé ;
- *interface* : la plate-forme est dotée d'une interface graphique permettant de représenter le diagnostic ainsi obtenu sous plusieurs formes exploitables par un opérateur de supervision.

Le fonctionnement de Ddyp se décompose en deux étapes.

1. Une phase *hors-ligne* : elle consiste à mettre en œuvre le modèle à exploiter, à le compiler afin d'établir une décentralisation de ce modèle et à construire les diagnostiqueurs issus de cette décentralisation.
2. Une phase *en-ligne* : elle se charge de recevoir les flux d'observations issus du système supervisé, de produire un diagnostic de ces flux par l'approche décentralisée, et de rafraîchir les informations de diagnostic visibles par l'opérateur de supervision.

[1] Prononcé dédyp.

6.2.1 Modélisation

L'objectif est de proposer un moyen simple de décrire un modèle à l'aide d'un *langage de description* qui sera l'unique point d'entrée de Ddyp. Nous nous sommes inspirés du langage Estelle bien connu dans le domaine de la spécification et la vérification de système [Turner 93] [ISO 97]. Ce choix a été guidé par plusieurs facteurs :

1. ces types de langages décrivent très intuivement des automates qui communiquent via des canaux de communications ; il nous est facile de traduire cette description dans des structures de données internes issues de notre formalisme (voir section 3.2.3.4 page 56) ;
2. ces langages offrent une représentation modulaire et hiérarchique : ils permettent donc de produire des modèles de façon simple, et d'utiliser des bibliothèques de composants pré-établies.

Le langage utilisé décrit le comportement du système supervisé avec une hiérarchie de *modules*. Un module existe sous deux formes :

1. les *modules élémentaires* : chaque module de ce type décrit un automate communicant, il représente le comportement d'un composant, en particulier les composants élémentaires ;
2. les *modules non-élémentaires* : chaque module non-élémentaire décrit un ensemble de modules fils ainsi que la façon dont ces modules communiquent entre eux.

La description du modèle est donc composé de deux étapes :

1. la spécification de l'ensemble des modèles comportementaux à l'aide des modules élémentaires ;
2. la construction du modèle structurel à l'aide de la hiérarchie de modules (non-élémentaires).

6.2.1.1 Description d'un composant élémentaire

Le comportement d'un composant élémentaire est décrit par un automate communicant dans un module élémentaire. Un événement est représenté par la présence d'un *message* sur un *port de communication*. S'il s'agit d'un événement issu de l'extérieur, alors il sera décrit par un message sur un *port d'entrée*. Si l'événement est produit par le composant élémentaire, alors il sera décrit par un message sur un *port de sortie*.

Exemple L'événement de panne primaire *ruptureCx12* (voir figure 3.7 page 62) sera décrit par le message *rupture* sur le port *panne* du module élémentaire représentant le composant *Cx12*. Le port *panne* est un port d'entrée. De même, l'événement *CM1_attenteCx12* sera décrit par le message *attente* sur le port de sortie *versCMO* du module élémentaire représentant le composant *Cx12* (*versCMO* signifiant *vers commutateur ouest*). Ce même événement sera également représenté par le message *attente* sur le port d'entrée *deCxE* du module élémentaire représentant le composant *CM1cnx* (*deCxE* signifiant *de la connexion Est*)

Un module élémentaire décrit donc les changements d'états du composant en fonction de la présence de messages sur des ports d'entrée ainsi que la réaction à ces changements par l'envoi de messages sur des ports de sortie.

Exemple Voici la description du module élémentaire associé à la connexion entre le commutateur 1 et le commutateur 2 de Toynet :

```
MODULE Cnx12;

IP
        INPUT panne : (rupture,rétablissement);
        OUTPUT versCMO: (attente,fin_attente);
        OUTPUT versCME: (attente,fin_attente);
END;

BEHAVIOR BodyCnx FOR Cnx;

        STATE z1,z2;
        INITIALIZE TO z1;

        TRANS
            FROM z1 TO z2
               WHEN panne.rupture
               OUTPUT versCMO.attente
               OUTPUT versCME.attente
        ;

        TRANS
            FROM z2 TO z1
               WHEN panne.rétablissement
               OUTPUT versCMO.fin_attente
               OUTPUT versCME.fin_attente
        ;
END;
```

Ce modèle traduit le fait qu'une connexion peut se rompre et se rétablir. Dans les deux cas, les commutateurs adjacents détectent cette rupture : cette détection est modélisée par l'envoi des messages *rupture* ou *rétablissement* sur les ports communiquant avec les commutateurs.

6.2.1.2 Description des communications entre composants élémentaires

Le modèle structurel décrit les communications entre composants élémentaires. La description de ce modèle est établie dans un module non-élémentaire et s'appuie sur :

- les *modules sous-jacents* ;
- les *points d'interfaces* (ports d'entrées et de sorties) ;
- les *connexions* ;
- les *attachements*.

Un *module non-élémentaire* décrit le modèle structurel associé à l'ensemble de ses modules fils. Deux modules fils communiquent entre eux à l'aide de *connexions*. Une connexion est une

association entre un port de sortie d'un module fils et un port d'entrée d'un autre module fils. Du fait de la hiérarchie des modules, certains ports de modules fils peuvent être connectés à des ports du module parent. Chaque port de ce type est lié à l'aide d'un *attachement*. Un tel attachement signifie que si le module fils émet ou reçoit un message sur l'un de ses ports attachés alors le module parent émet ou reçoit ce même message sur le port attaché correspondant. Contrairement à une connexion, un attachement associe uniquement deux ports du même type.

Exemple La figure 6.1 présente la description du modèle structurel associé aux deux composants élémentaires *CM1ctl* et *CM1cnx* (voir section 3.2.3.1 page 53). Ce module non-élémentaire représente le commutateur *CM1* complet.

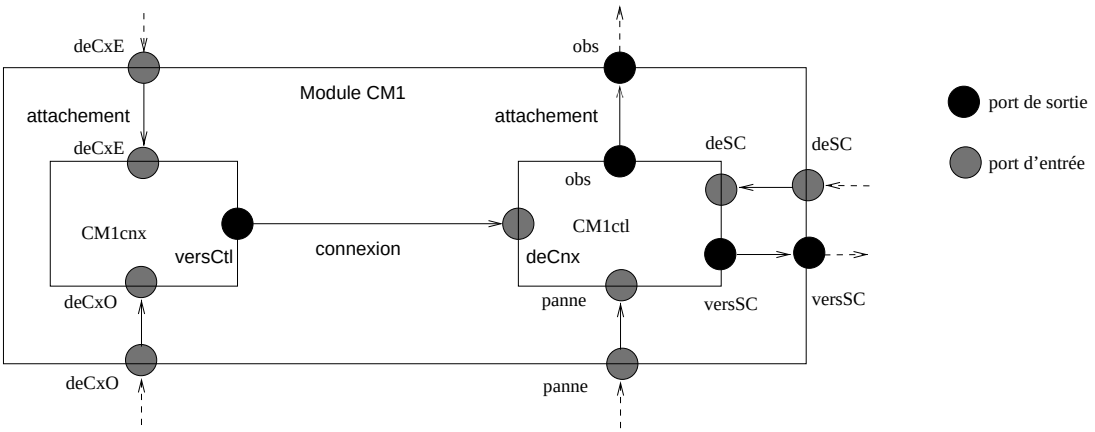

FIG. 6.1 – Modèle non-élémentaire représentant le commutateur *CM1* de Toynet.

Ce module est établi à l'aide de la description suivante :

```
MODULE CM1;

IP
        INPUT deCxE: (chg);
        INPUT deCxO: (chg);
        INPUT panne: (bloque,fin_reinit);
        OBSERVABLE obs: (CM1op,CM1blc,SC1op,cx12,cx31);
        INPUT deSC: (opérationnel,reinit);
        OUTPUT versSC: (a_relancer);
END;

BEHAVIOR BodyCM1 FOR CM1;

   MODULE CMctl;
   ...
   END;
```

```
            MODULE CMcnx;
            ...
            END;

        END;

        STRUCTURE
                ATTACH deCxO TO CMcnx.deCxO;
                ATTACH deCxE TO CMcnx.deCxE;
                ATTACH panne TO CMctl.panne;
                ATTACH deSC TO CMctl.deSC;
                ATTACH versSC TO CMctl.versSC;
                ATTACH obs TO CMctl.obs;

                CONNECT CMcnx.versCtl TO CMctl.deCnx;
        END;
END;
```

Les ports de sortie qui sont déclarés `OBSERVABLE` sont les ports associés aux événements observables. Tout message associé à tel port correspond à un événement observable.

Dans Toynet, on considère que les communications entre les différents composants élémentaires sont instantanées. Néanmoins, si ce n'était pas le cas, on peut introduire une notion de latence sur les connexions, en représentant par exemple une connexion par une file bornée (voir section 3.2.3.2 54). Dans ce cas, le langage permet de déclarer de telle politique de communication. Imaginons un instant que la connexion entre *CM1ctl* et *CM1cnx* soit représentée par une file de 4 messages, dans ce cas, il suffit de déclarer le port de sortie de *CM1cnx* vers *CM1ctl* par :

```
versCtl:{chg} QUEUE[4];
```

6.2.1.3 Choix de la décentralisation

Une fois l'étape de modélisation effectuée, le modèle contient un ensemble de modules hiérarchiquement organisés. Chaque module élémentaire correspond à un composant élémentaire. La deuxième étape du processus hors-ligne est d'établir la décentralisation du modèle. Cette décentralisation consiste à partitionner l'ensemble des modules élémentaires et à compiler un diagnostiqueur local associé à chaque élément de la partition. Ddyp autorise n'importe quelle décentralisation, c'est à l'utilisateur de la définir en fonction du modèle et des critères fondés sur l'efficacité de l'approche (voir en particulier la section 4.3.6.3 à ce sujet). Le résultat de la décentralisation est un ensemble de diagnostiqueurs locaux exploitables en-ligne pour l'établissement de diagnostics locaux.

6.2.2 Architecture de diagnostic

La plate-forme pour le diagnostic en-ligne est une application distribuée qui met en œuvre l'approche décentralisée du diagnostic. Étant donné l'ensemble des diagnostiqueurs locaux compilés dans la phase hors-ligne, le déploiement de la plate-forme est un ensemble d'objets communicants via un bus Corba suivant le paradigme du client/serveur [Group 01]. La figure 6.2 présente ce déploiement d'objets Corba. L'avantage d'une telle architecture est qu'elle est très facilement déployable sur un système informatique dédié à la supervison. De plus, elle offre la possibilité de connecter des interfaces graphiques dédiées à la supervison d'un système donné.

Diagnostiqueur Un objet diagnostiqueur correspond au déploiement d'un ou de plusieurs processus de calcul de diagnostics locaux [2]. Le nombre d'objets diagnostiqueur dépend de la décentralisation du système. Ils sont instanciés à partir du résultat de la décentralisation. Ils dépendent aussi de la topologie du système d'observation. L'idéal est d'instancier un diagnostiqueur « au plus près » de la sortie du flux d'observations afin d'avoir des propriétés les plus fines possibles sur l'ordre de réception des observations. La deuxième contrainte pour le déploiement d'un tel objet est qu'il existe un moyen de communication sûr entre cet objet diagnostiqueur et le reste de la plate-forme.

Fusionneur Cet objet fait parti du système de coordination de la plate-forme de diagnostic. Son objectif est d'appliquer l'opération de fusion sur les diagnostics (opération $\odot$ définie par l'algorithme 7 page 114). S'il existe plusieurs instances de ce type d'objets, il sera possible d'appliquer des fusions en parallèle. Le nombre de fusionneurs dépend du nombre de diagnostics locaux à fusionner dans une fenêtre temporelle et des ressources informatiques distribuées disponibles.

Coordinateur Le coordinateur a pour objectif d'appliquer la stratégie de fusion (voir section 4.5 page 115) et de gérer les fenêtres temporelles. À la fin de chaque fenêtre temporelle, il récupère les interactions proposées par les différents diagnostics locaux lors de cette fenêtre et commande les fusions à effectuer en fonction de la stratégie calculée. Pour le moment, le coordinateur ne gère que les fenêtres temporelles considérées comme sûres (voir section 5.3 page 127).

Interface L'interface est l'objet qui centralise les résultats. En particulier, il a la charge de stocker les diagnostics associés aux différentes fenêtres temporelles et de proposer des abstractions sur ces résultats qui soient exploitables par un opérateur de supervision. En particulier, cet objet a pour objectif de proposer une interface de programmation permettant de brancher des interfaces graphiques dédiées à la supervision d'un système donné.

[2] Attention, un objet diagnostiqueur ne correspond pas forcément au déploiement d'une structure diagnostiqueur mais éventuellement de plusieurs (processus multi-threadé).

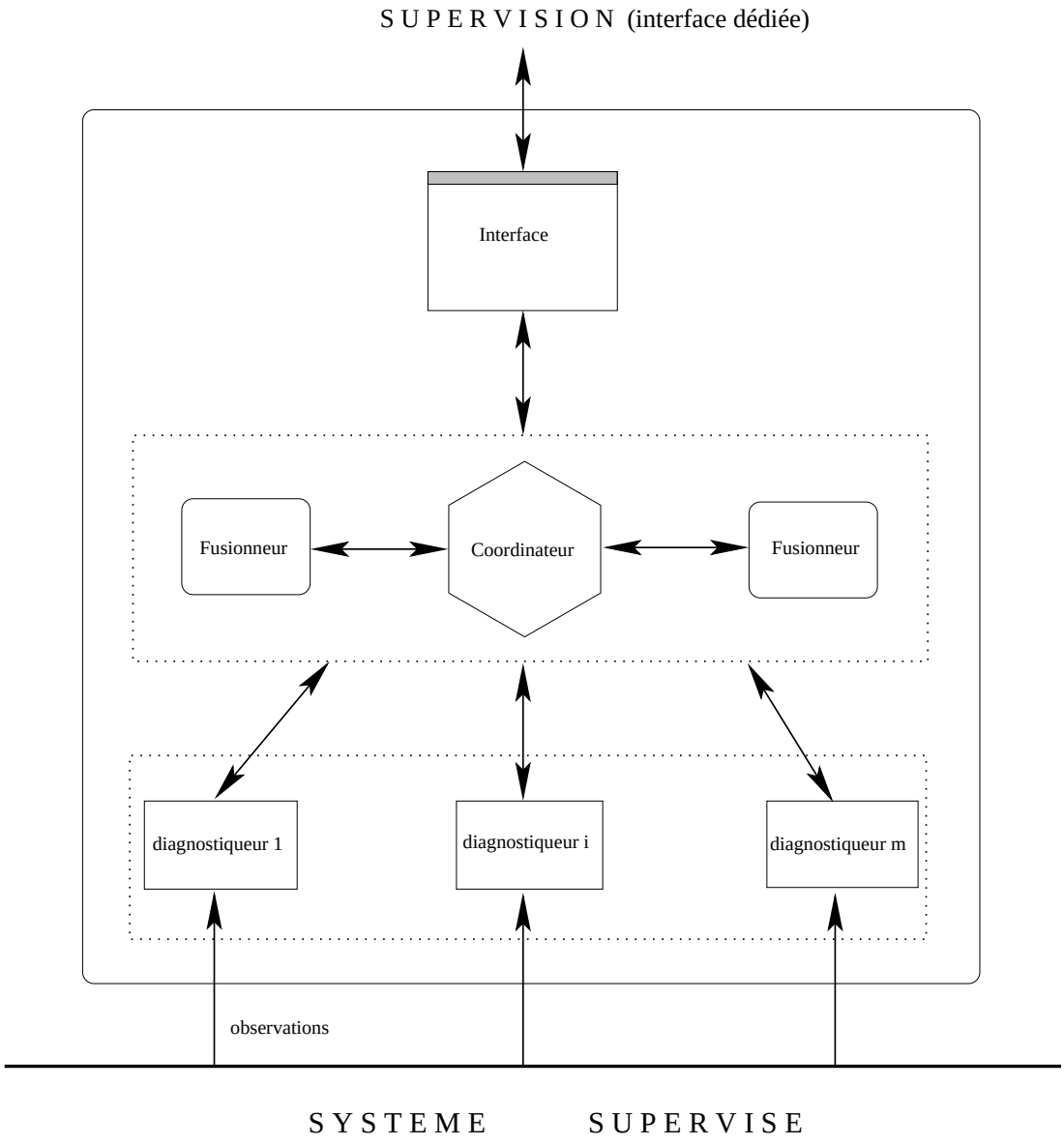

FIG. 6.2 – Déploiement de la plate-forme Ddyp.

6.2.3 Interface vers l'opérateur

Au dessus de la plate-forme Ddyp, nous avons également mis en œuvre une interface graphique permettant de contrôler la plate-forme et de visualiser sous différentes formes le résultat du diagnostic (voir figure 6.3). Cette interface a la particularité d'être générique, elle ne dépend aucunement du réseau supervisé. Elle peut être utilisée en-ligne ou hors-ligne (chargement de diagnostic à analyser).

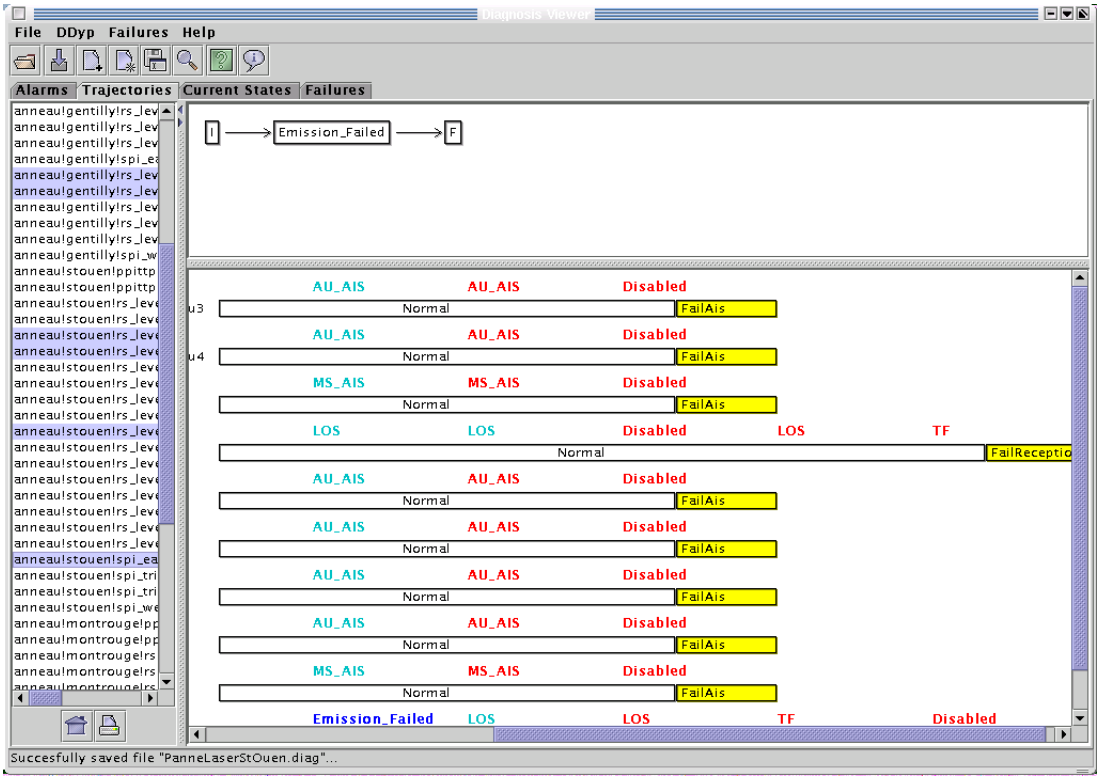

FIG. 6.3 – Interface graphique de Ddyp.

Cette interface offre différentes vues du diagnostic. Une première vue dite *corrélation d'alarmes* permet de lister les alarmes reçues. Un clic sur une alarme permet de mettre en valeur les alarmes de la liste qui sont en corrélation avec celle qui est sélectionnée (cette interface offre une vision du problème posé par exemple dans [Jakobson et Weissman 93] (voir section 2.3.2.2). Une autre vision possible est la vision statistique, qui consiste à afficher des certitudes sur l'apparition de telle ou telle panne, son taux d'occurrence... La troisième offre un moyen d'explorer les différentes explications des observations. Il suffit pour cela de sélectionner un ensemble de composants élémentaires, l'interface propose alors un moyen de dérouler les événements qui ont pu avoir lieu sur les composants sélectionnés et qui expliquent les observations.

6.2.4 Bilan

Ddyp est une plate-forme qui met en œuvre toute la chaîne de tâches nécessaires à la mise en place d'un système de diagnostic décentralisé. Ces tâches vont de la production d'un modèle à l'aide d'un langage de description intuitif au calcul en-ligne d'un diagnostic et à son analyse (en-ligne ou hors-ligne). Cette plate-forme nous a permis de valider l'approche décentralisée sur des cas concrets de réseaux de rélécommunications.

6.3 Étude sur le réseau Transpac

6.3.1 Introduction

Cette étude a été effectuée en rapport avec les travaux issus du projet Gaspar (voir section 2.4.2.5). Ce projet a permis entre autres d'établir un modèle de comportement en cas de pannes [Rozé 97a] du réseau Transpac. En s'appuyant sur ce modèle, nous présentons une étude du comportement de notre système de diagnostic sur une partie de ce réseau réel.

Ce sous-réseau à commutation de paquets est composé de 8 commutateurs gérés par deux centres techniques. Chaque commutateur est associé avec 4 stations. Les stations sont de deux types :

1. les stations d'exploitation (STE) ;
2. les stations de gestion (STG).

Les stations d'exploitations sont utilisées pour le fonctionnement basique du commutateur. Chaque commutateur dispose deux stations de ce type : la station primaire (STE1) sert dans le cadre du fonctionnement nominal du commutateur et la station secondaire (STE2) est la station de secours. La présence de deux stations augmentent la robustesse du réseau en cas de panne, en effet, si STE1 tombe en panne, un mécanisme permute le contexte (on parle aussi de basculement de rôle) afin que la station STE2 prenne le relais. On dit ainsi que l'ensemble des deux stations STE1 et STE2 constituent l'unité d'exploitation (UE) d'un commutateur.

Les stations de gestion sont utilisées pour contrôler le routage du commutateur. Comme pour les stations d'exploitation, ces stations constituent une unité de gestion (USG). Par le même mécanisme, les deux stations de gestion STG1 et STG2 sont respectivement les stations primaire et secondaire (de secours) de l'unité de gestion.

La topologie du réseau ainsi étudié est présentée sur la figure 6.4. Le réseau ainsi décrit est constitué de 42 équipements.

Dans cette architecture, les alarmes émises par les différentes stations le sont par l'intermédiaire du commutateur associé. Cela implique en particulier, que si ce commutateur est en état de dysfonctionnement, les alarmes émises par les stations sont perdues : ce problème montre que le phénomène de masquage est très présent dans cette architecture.

6.3.2 Comportements des équipements

Un centre technique peut émettre deux types d'alarmes : *cvhs* et *cves*. L'alarme *cvhs* est émise lorsque le centre technique tombe en panne : cette panne est soit une réinitialisation

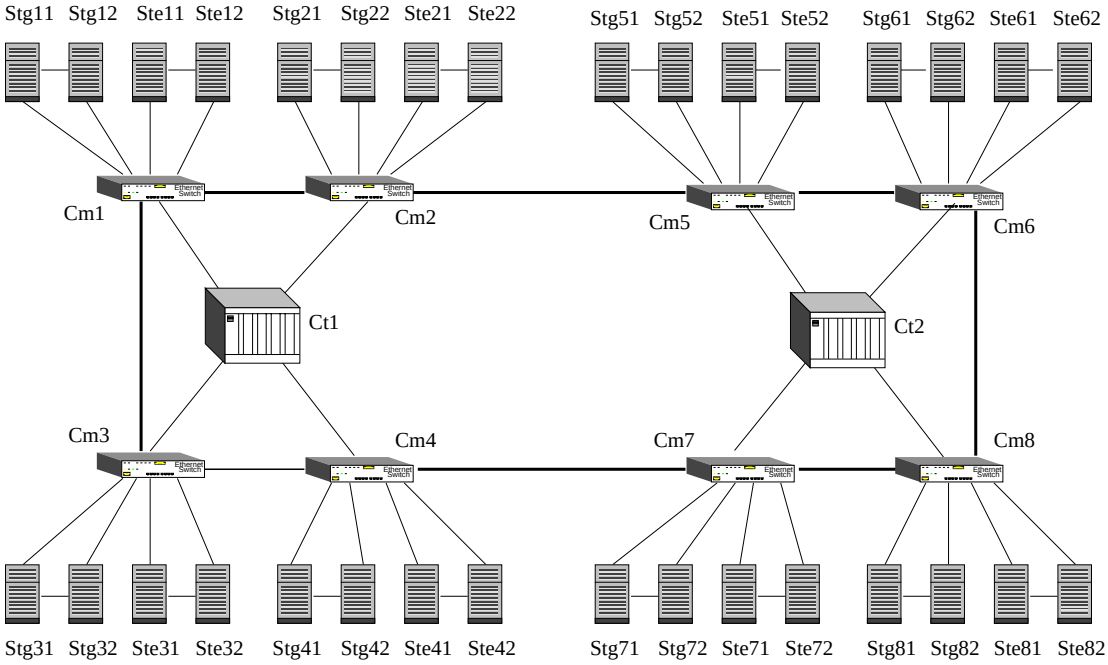

FIG. 6.4 – Topologie du réseau étudié.

(*reboot*), soit une rupture de liaison (*cut*). L'alarme *cves* est émise lorsque le centre technique retrouve un fonctionnement normal (fin de la réinitialisation, recouvrement de la liaison).

Concernant les commutateurs, ils peuvent aussi émettre deux types d'alarmes : *n003* et *n004*. L'alarme *n003* est émise lorsque le commutateur se bloque ou s'arrête (*blk*) ou bien lorsque le commutateur se réinitialise.

Pour les stations (de gestion ou d'exploitation), les alarmes émises sont du type *p089* muni de paramètres identifiant le type de la station, le fait qu'il s'agit d'une station primaire ou non. Ces alarmes sont émises dans plusieurs cas.

La figure 6.5 présente le modèle d'une station primaire de gestion (STG1). Dans ce cas par exemple, l'état d'une station peut être *actif*, *réinitialisation* et *arrêt*. De façon indépendante, cette station peut être masquée ce qui fait découler trois autres états *actifM*, *réinitialisationM* et *arretM*. Les événements de pannes exogènes pouvant se produire sur une telle station sont deux 2 types. Le premier est l'arrêt de la station (*blcStg1*), le deuxième est la réinitialisation (*reinitStg1*). Ces pannes sont intermittentes, des événements de retour leur correspondent : *retourStg1* représente la fin de l'arrêt, *finReinitStg1* représente la fin de la réinitialisation de la station.

Une station de gestion primaire peut interagir avec le commutateur qui lui est associé ainsi qu'avec la station secondaire de l'unité de gestion (voir figure 6.6). Par exemple dans le cas où la station primaire se bloque, un basculement entre la station primaire et la station secondaire se produit, ce basculement est modélisé par l'événement *bascStg2* émis par la station primaire et reçu par la station secondaire. De même, lorsque la station primaire retrouve son état de fonctionnement nominal, elle reprend la main et libère la station secondaire : cette libération est

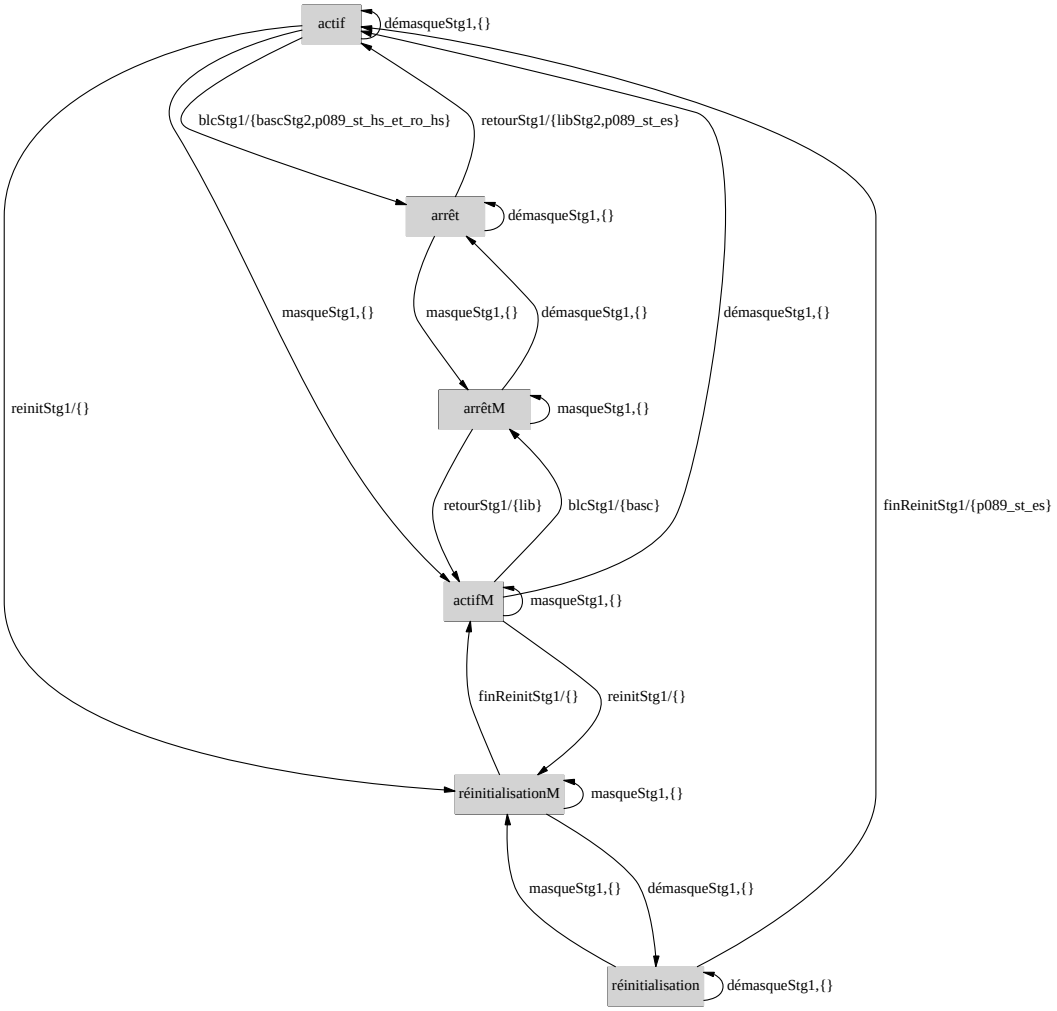

FIG. 6.5 – Modèle d'une station de gestion primaire *Stg1*.

modélisée par l'émission d'un événement *libStg2* par la station primaire vers la station secondaire. Quant au phénomène de masquage, il est représenté à l'aide des événements *masqueStg1* et *démasqueStg1*. Ces événements sont émis par le commutateur associé. Lorsque la station est dans un état masqué, aucune alarme n'est produite.

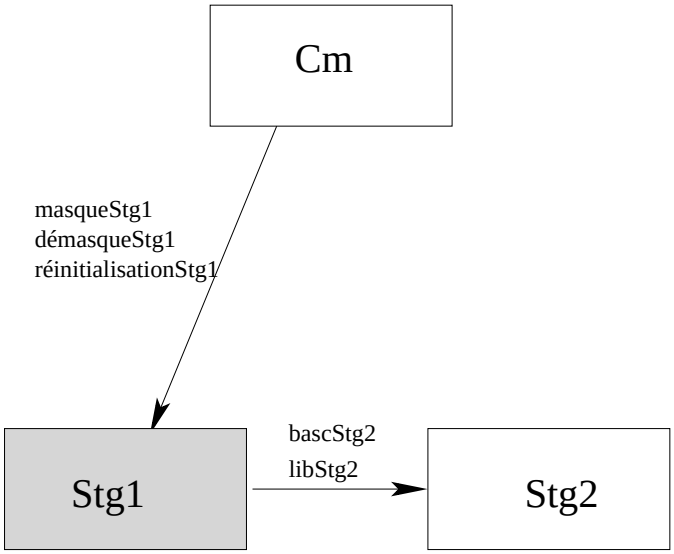

FIG. 6.6 – Interactions d'une station STG1 avec son voisinage.

Dans le diagnostic de ce réseau, l'une des difficultés vient du fait que l'observation d'une alarme ne suffit pas en général à identifier une panne. Par exemple, l'alarme *n003* peut correspondre soit à l'arrêt d'un commutateur, soit à sa réinitialisation. Afin de discriminer ces deux hypothèses, il est nécessaire de vérifier les observations sur les autres équipements car la propagation des deux pannes du commutateur n'est pas la même et provoque dans la majeure partie des cas une signature observée différente.

6.3.3 Résultats de l'étude

Dans cette expérimentation, nous avons calculé le diagnostic global pour une fenêtre temporelle contenant 56 alarmes (voir tableau 6.1) émises par 20 composants du réseau présenté dans la section précédente.

La plate-forme de diagnostic dispose de trois fusionneurs afin de paralléliser les calculs. En ce qui concerne le calcul des diagnostics locaux, on a mis en place un diagnostiqueur par composant (d'où 42 diagnostiqueurs locaux).

6.3.3.1 Analyse du résultat

Le temps de calcul de chaque diagnostic local sur la fenêtre présentée dans le tableau 6.1 est inférieur à 100ms, ce qui est efficace.

Ste11	*Ste12*	*Stg21*	*Stg22*
p089_2_st_hs *p089_2_st_es* *p089_2_ro_es* *p089_2_ro_hs* *p089_1_ro_es*	*p089_2_ro_es* *p089_1_ro_es* *p089_2_st_hs*	*p089_ro_hs* *p089_st_hs* *p089_st_es*	*p089_sec_st_hs* *p089_nom_st_es* *p089_nom_ro_rope* *p089_nom_st_hs* *p089_nom_ro_nop* *p089_sec_st_es* *p089_sec_st_hs* *p089_sec_ro_hs*
Cm4	*Stg41*	*Stg42*	*Ste41*
n003 *n004*	*p089_st_es*	*p089_sec_st_hs* *p089_sec_ro_hs*	*p089_1_ro_es*
Ste42	*Cm5*	*Ste51*	*Ste52*
p089_2_st_es	*n003* *n004*	*p089_2_st_hs* *p089_2_st_es* *p089_2_ro_es* *p089_2_ro_hs* *p089_1_ro_es*	*p089_2_ro_hs* *p089_1_ro_es* *p089_2_st_hs*
Cm6	*Stg61*	*Stg62*	*Cm8*
n003 *n004*	*p089_st_hs* *p089_ro_hs* *p089_st_es*	*p089_sec_st_hs* *p089_nom_st_es* *p089_nom_ro_rope* *p089_nom_st_hs* *p089_nom_ro_nop* *p089_sec_st_es* *p089_sec_st_hs* *p089_sec_ro_hs*	*n003* *n004*
Stg81	*Stg82*	*Ste81*	*Ste82*
p089_st_es	*p089_sec_st_hs* *p089_sec_ro_hs*	*p089_1_ro_es*	*p089_2_st_es*

TAB. 6.1 – L'ensemble des séquences d'alarmes observé durant une fenêtre temporelle.

D'après les alarmes reçues, aucune interaction n'est possible entre les centres techniques *Ct1, Ct2* et leurs commutateurs. Néanmoins, les diagnostics locaux des commutateurs supposent que de telles interactions sont possibles : en effet, chaque diagnostic local de commutateur émet l'hypothèse qu'il y a eu masquage (d'où l'envoi par les centres techniques d'événements du type *masque* et *démasque*). Après l'élimination des trajectoires incompatibles, les diagnostics locaux des commutateurs étant épurés, ces hypothèses d'interactions ont été éliminées. Ces diagnostics locaux sont devenus indépendants des diagnostics locaux des centres techniques : la fusion d'un diagnostic local de commutateur avec celui d'un centre technique est donc devenue inutile. C'est pour les mêmes raisons que les diagnostics locaux $\Delta_{Stg71}, \Delta_{Stg72}, \Delta_{Ste71}, \Delta_{Ste72}$ sont indépendants du diagnostic local Δ_{Cm7} et ne sont pas fusionnés. Le résultat du calcul du diagnostic sur cette fenêtre temporelle est ainsi constitué de 15 diagnostics indépendants (voir tableau 6.2). Ce résultat a été obtenu en 8 secondes (ce temps correspond au temps réel entre la première réception d'alarmes de la fenêtre et l'obtention du diagnostic global).

Ce tableau présente également les états possibles du système à la fin de la fenêtre temporelle [3]. On voit en particulier les effets du masquage. Dans le diagnostic Δ_8 par exemple, les stations peuvent se trouver dans n'importe quel état non masqué. Ce diagnostic correspond au fait qu'au cours de la fenêtre temporelle, ces stations ont été masquées et que tout événement de panne a pu se produire pendant cette période de masquage. Lorsque le commutateur associé est retourné dans un état de bon fonctionnement, les stations ne sont plus masquées mais peuvent être dans n'importe quel état de panne. Seules des alarmes d'une fenêtre temporelle future sont en mesure de discriminer entre tel état de panne et un autre.

6.3.3.2 Comparaisons avec d'autres stratégies

Afin de montrer l'efficacité de la stratégie de fusion appliquée, nous l'avons comparé avec d'autres stratégies de fusions possibles. Cette comparaison de performance a été établie en mesurant les temps de calculs du diagnostic global fondé sur un sous-ensemble du réseau étudié. Ce sous-ensemble est constitué des composants *Ct2, Cm8, Stg81, Stg82, Ste81, Ste82*. Ces mesures ont été réalisées en considérant que les alarmes à diagnostiquer sont celles issues de ces 6 composants et présentées dans le tableau 6.1. Dans cette expérimentation, nous ne considérons pas le parallélisme, les mesures ont été effectuées en utilisant qu'un unique fusionneur. Les temps de calcul sont présentés sur la figure 6.7. Pour chaque étape de fusion (ici 5 étapes), la figure présente le temps de calcul cumulatif depuis la première étape.

La stratégie 1 est celle qui est utilisée par notre outil de diagnostic (voir section 4.5). Afin de mieux comparer les stratégies, les diagnostics indépendants (à savoir $\Delta_{Cm8,Stg81,Stg82,Ste81,Ste82}$ et Δ_{Ct2}) ont été fusionnés (cela constitue la dernière étape de fusion pour la stratégie 1). Nous obtenons en résultat le diagnostic global de ce sous-ensemble de composants de manière explicite. La deuxième stratégie étudiée applique elle aussi l'élimination des hypothèses locales impossibles avant d'effectuer les fusions. Par contre, cette stratégie ne privilégie pas la fusion des diagnostics interagissants. En particulier, lors de l'étape 2, les diagnostics fusionnés n'interagissent pas directement. La fusion à l'étape 2 ne valide ou n'invalide aucune hypothèse de diagnostic. Nous avons également expérimenté d'autres

[3] Dans cette expérience, nous considérons que la fenêtre temporelle est sûre.

Numéro	Diagnostic indépendant	États diagnostiqués
Δ_1	Δ_{Ct1}	*Ct1 :* {*actif*}
Δ_2	$\Delta_{Cm1,Stg11,Stg12,Ste11,Ste12}$	*Cm1 :* {*actif*}, *Stg11 :* {*actif*}, *Stg12 :* {*passif*}, *Ste11 :* {*actif*}, *Ste12 :* {*arret*}
Δ_3	$\Delta_{Cm2,Stg21,Stg22,Ste22}$	*Cm2 :* {*actif*}, *Stg21 :* {*actif*}, *Stg22 :* {*arret*}, *Ste22 :* {*passif*}
Δ_4	Δ_{Ste22}	*Ste22 :* {*passif*}
Δ_5	$\Delta_{Cm3,Stg31,Stg32,Ste31,Ste32}$	*Cm3 :* {*actif*}, *Stg31 :* {*actif*}, *Stg32 :* {*passif*}, *Ste31 :* {*actif*}, *Ste32 :* {*passif*}
Δ_6	$\Delta_{Cm4,Stg41,Stg42,Ste41,Ste42}$	*Cm4 :* {*actif*}, *Stg41 :* {*actif*}, *Stg42 :* {*arret,passif*}, *Ste41 :* {*actif*}, *Ste42 :* {*passif*}
Δ_7	Δ_{Ct2}	*Ct2 :* {*actif*}
Δ_8	$\Delta_{Cm5,Stg51,Stg52,Ste51,Ste52}$	*Cm5 :* {*actif*}, *Stg51 :* {*actif,arret,réinitialisation*}, *Stg52 :* {*actif, arret, passif, réinitialisation*}, *Ste51 :* {*actif, arret, passif, réinitialisation*}, *Ste52 :* {*actif, arret, passif, réinitialisation*}
Δ_9	$\Delta_{Cm6,Stg61,Stg62,Ste61,Ste62}$	*Cm6 :* {*actif*}, *Stg61 :* {*actif, arret, réinitialisation*}, *Stg62 :* {*actif,arret,passif*}, *Ste61 :* {*actif, arret, passif, réinitialisation*}, *Ste62 :* {*actif, arret, passif, réinitialisation*}
Δ_{10}	Δ_{Cm7}	*Cm7 :* {*actif*}
Δ_{11}	Δ_{Stg71}	*Stg71 :* {*actif*}
Δ_{12}	Δ_{Stg72}	*Stg72 :* {*passif*}
Δ_{13}	Δ_{Ste71}	*Ste71 :* {*actif*}
Δ_{14}	Δ_{Ste72}	*Ste72 :* {*passif*}
Δ_{15}	$\Delta_{Cm8,Stg81,Stg82,Ste81,Ste82}$	*Cm8 :* {*actif*}, *Stg81 :* {*actif*}, *Stg82 :* {*arret,passif*}, *Ste81 :* {*actif*}, *Ste82 :* {*passif*}

TAB. 6.2 – Résultat du diagnostic : un ensemble de 15 diagnostics indépendants

stratégies dans lesquelles on n'utilisait pas l'élimination des hypothèses locales impossibles. Dans ce cas cette élimination est effectuée durant la fusion mais de manière très inefficace : le temps de calcul nécessaire est dans ce cas de plusieurs minutes !

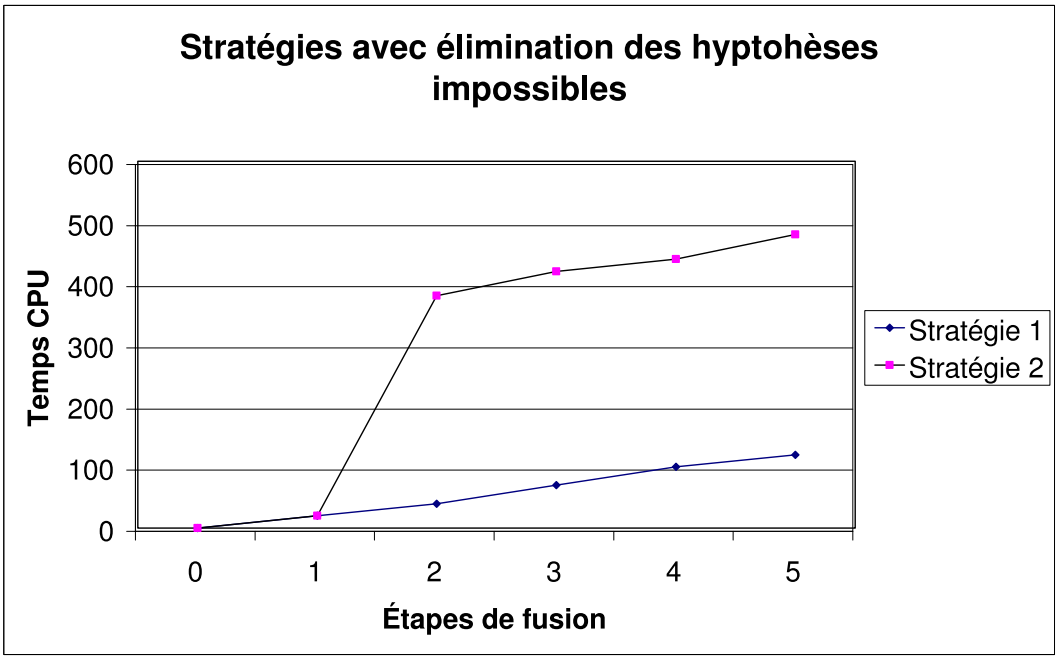

FIG. 6.7 – Comparaison des performances entre deux stratégies de fusion (temps en ms).

6.4 Étude sur un réseau SDH

6.4.1 Introduction

Notre deuxième cas d'étude est issu d'un projet RNRT (Réseau National de Recherche en Télécommunications) : le projet Magda (Modélisation et Apprentissage pour une Gestion Distribuée des Alarmes).

Le réseau étudié est constitué de quatres multiplexeurs SDH (hiérarchie numérique synchrone) formant un réseau en forme d'anneau (voir figure 6.8). Nous présentons dans cette section, l'expérimentation qui a été mise en place pour la revue finale du projet Magda.

Chaque multiplexeur ADM (*Add and Drop Multiplexer*) est situé dans une ville différente en Ile-de-France. Tous les multiplexeurs excepté celui d'Aubervilliers proposent des connexions vers des clients (les connexions de type PDH ou STM).

Le réseau de gestion associé à cet anneau (le RGT, voir section 1.3) est constitué d'un ensemble d'objets gérés associés à la norme SDH [Bouyer 97, UT 92a, UT 94, UT 96]. La figure 6.9 présente les objets gérés associés au multiplexeur de Montrouge.

Les objets sont classés en couches hiérarchiques, de la couche physique SPI (*Synchronous Physical Interface*) aux couches de plus haut niveau HOP (*High Order Path*) et LOP (*Low*

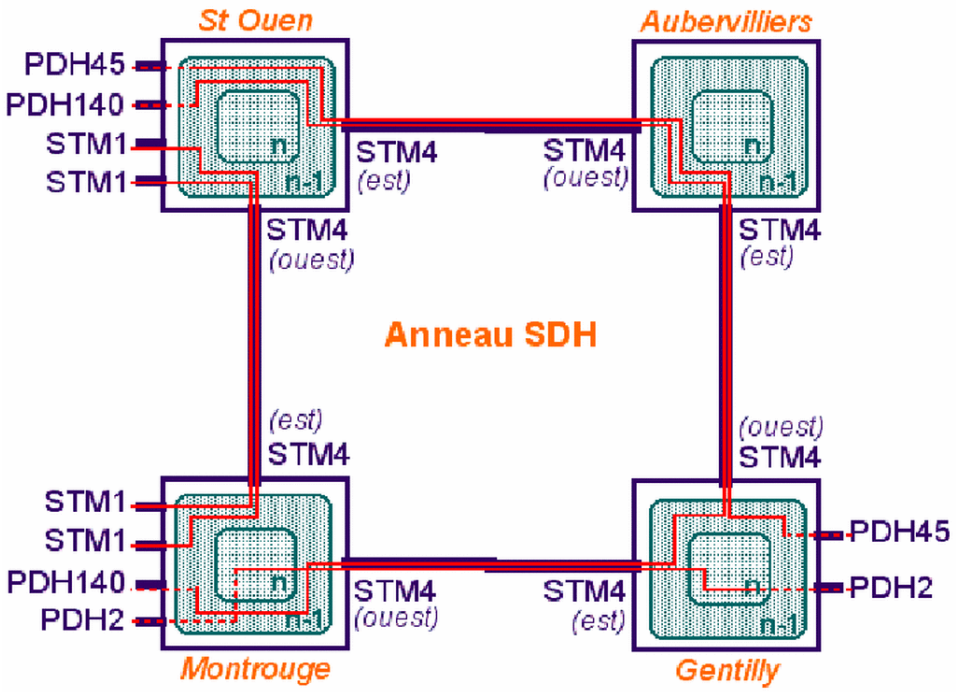

FIG. 6.8 – Topologie du réseau SDH étudié.

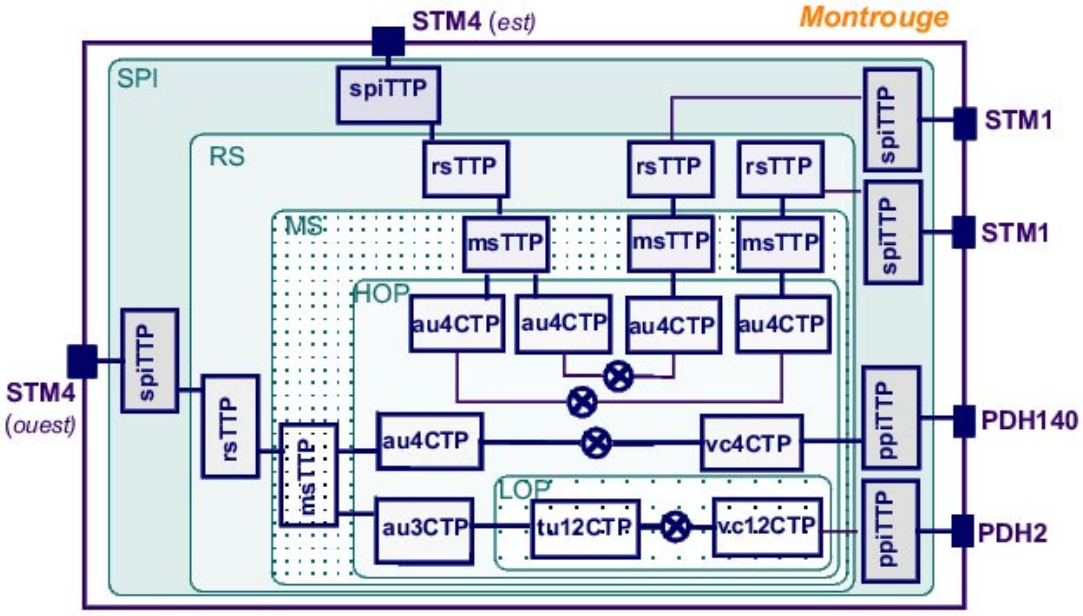

FIG. 6.9 – Objets gérés associés au multiplexeur de Montrouge.

Order Path). Chaque objet peut émettre des alarmes suite à des pannes pouvant se produire sur l'objet géré en question. Ces alarmes peuvent être aussi issues de la réception de messages sur l'objet en question, messages provenant d'un autre objet géré. En particulier, si un objet est sujet à un dysfonctionnement, il envoie via le réseau un message à l'objet dual du site voisin.

Exemple Si l'objet `msTTP` de la connexion *ouest* de Montrouge est sujet à un dysfonctionnement, il va envoyer un message `MS-AIS` à l'objet `msTTP` de la connexion *est* de Gentilly (voir figure 6.8). Ce message traversera la couche RS puis SPI de Montrouge puis les couches SPI, RS de Gentilly avant d'atteindre l'objet destination. Si sur ce chemin, un des objets est en dysfonctionnement, le message est perdu.

6.4.2 Modélisation

La modélisation de ce réseau a été effectuée dans le cadre du projet Magda par d'autres partenaires. Le modèle réalisé a été établi à partir des normes SDH [UT 92a, UT 94, UT 96] et d'une expertise effectuée par le superviseur de ce réseau. Le formalisme utilisé est un formalisme de pièce dont la notation est présentée sur la figure 6.10.

Cette pièce définit un comportement basique d'un objet géré. Elle informe que si une certaine *précondition* est vérifiée sur l'objet, si un certain *message* arrive sur cet objet alors, selon des *conditions*, des messages et des alarmes sont générés. L'objet passe alors dans un état

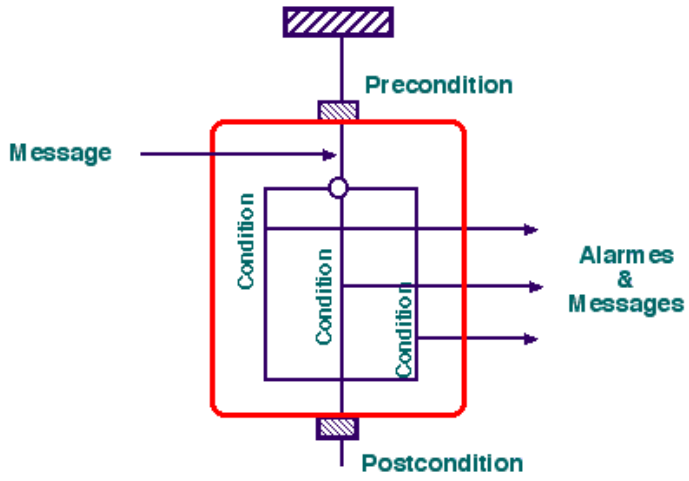

FIG. 6.10 – Définition d'une pièce.

vérifiant une certaine *postcondition*.

6.4.2.1 Acquisition du modèle dans Ddyp

Le modèle ainsi défini a été traduit dans le langage de description de Ddyp. Le composant élémentaire correspond à un objet géré et représente l'ensemble de ces comportements possibles. Le principe de la traduction est le suivant :

- on considère la *précondition*, pour chaque *condition*, on définit un état $E1$ du composant élémentaire représentant l'assertion *précondition* $\wedge$ *condition* ;
- on considère la *postcondition* et pour chaque *condition* activable à partir de cette *postcondition*, on définit l'état $E2$ du composant élémentaire représentant l'assertion *postcondition* $\wedge$ *condition* ;
- la pièce est ainsi traduite par un ensemble de transitions décrites comme suit :

```
TRANS
  FROM E1 TO E2
    WHEN portEntree.message
    OUTPUT portSortie.alarmes
    OUTPUT portSortie2.messages
;
```

Au niveau du modèle structurel, la vision hiérarchique que propose le langage de Ddyp facilite sa réalisation. La hiérarchie de modules associée au multiplexeur de Montrouge correspond par exemple à celle présentée sur la figure 6.9. Au niveau de la communication des messages, nous avons considéré que les connexions entre les différents objets gérés étaient représentables par des files de capacité 1.

6.4.2.2 Décentralisation du modèle

Le modèle de l'anneau SDH est constitué de 72 composants élementaires. La décentralisation du modèle qui a été choisie est en fonction des sites. Voici le nombre de grappes de composants élémentaires en fonction des sites :

- Montrouge : 3 grappes ;
- St Ouen : 3 grappes ;
- Aubervilliers : 2 grappes ;
- Gentilly : 2 grappes.

La figure 6.11 présente les 3 grappes résultat de la décentralisation du site de Montrouge. Cette décentralisation a été établie en fonction de différents paramètres :

1. le *nombre de composants élémentaires* : chaque diagnostiqueur local se charge d'un nombre de composants semblable à chacun des autres ;
2. les *interactions entre composants* : une grappe est constituée d'un ensemble de composants élémentaires qui communiquent ensemble, on évite ainsi de compiler le comportement de composants concurrents.

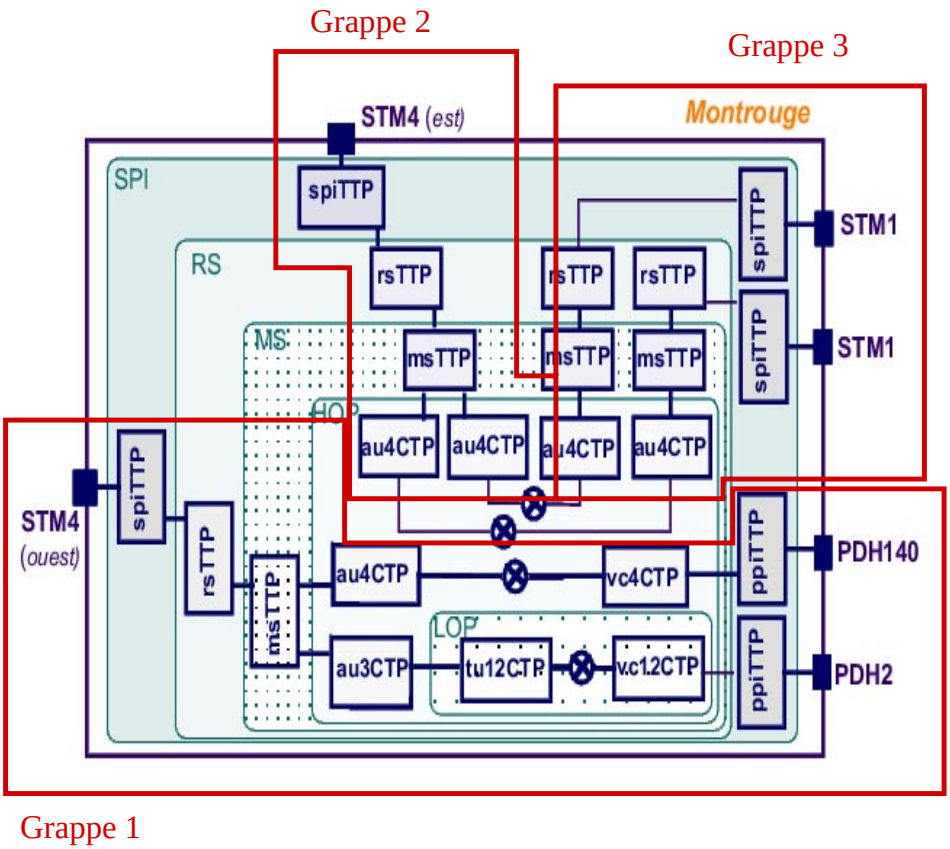

FIG. 6.11 – Décentralisation du site de Montrouge.

6.4.3 Diagnostic

6.4.3.1 Observabilité du système

Dans cette expérimentation, nous avons considéré que le réseau était supervisé par un seul centre de supervision. Ce centre de supervision est muni d'un capteur qui reçoit et date toutes les alarmes. Étant donnée la topologie du système, nous considérons qu'il existe un canal de communication FIFO entre un site et le capteur du superviseur. Chaque canal de communication est considéré indépendant des autres. Autrement dit, pour toute alarme a_1 reçue avant a_2, si a_1 et a_2 proviennent du même site, on considère que $a_1 \preceq a_2$.

6.4.3.2 Déploiement de Ddyp

Pour cette expérimentation nous disposions de deux ordinateurs portables. Le déploiement de Ddyp sur ces deux machines est le suivant.

- L'ensemble des 8 diagnostiqueurs locaux : tous les diagnostiqueurs d'un même site sur la même machine.
- Deux fusionneurs : un par machine afin de profiter du parallélisme. Ce choix de deux fusionneurs est aussi guidé par la topologie du réseau : on est en effet assuré de toujours avoir en résultat au moins deux diagnostics indépendants, puisqu'il existe dans ce réseau deux sous-ensembles de composants élémentaires qui ne communiquent jamais (voir sur le site de Montrouge figure 6.9).

6.4.3.3 Mise en place de la chaîne de diagnostic

Ddyp est le noyau d'une *chaîne de diagnostic* qui a été mise en place lors de la revue finale du projet Magda. Cette chaîne est constituée des élements suivants :

1. un *gestionnaire de réseau* : il a la charge de récupérer les alarmes du réseau et de les dater ;
2. Ddyp : il établit un diagnostic en fonction des alarmes récupérées par le gestionnaire ;
3. une *interface graphique d'exploitation* : il s'agit d'une interface graphique qui présente la topologie du réseau et les propagations de pannes diagnostiquées par Ddyp (figure 6.12).

Les communications entre les différents maillons de la chaîne sont effectuées *via* un bus Corba. Le gestionnaire de réseau est un module fourni par un partenaire industriel : la société Alcatel. Quant à l'interface graphique présentant la topologie, elle a été développée par un deuxième partenaire industriel : la société Ilog.

6.4.3.4 Interface graphique

L'interface graphique proposée par Ilog permet de présenter à l'opérateur le résultat du diagnostic de manière topologique. Par un jeu de couleurs sur les objets gérés, l'interface affiche une hypothèse de diagnostic, c'est-à-dire la présence de pannes primaires sur certains de ces objets ainsi que leur propagation respective. Cet affichage a l'intérêt d'être plus ergonomique pour les opérateurs de supervision. Les informations de diagnostic sont établies à

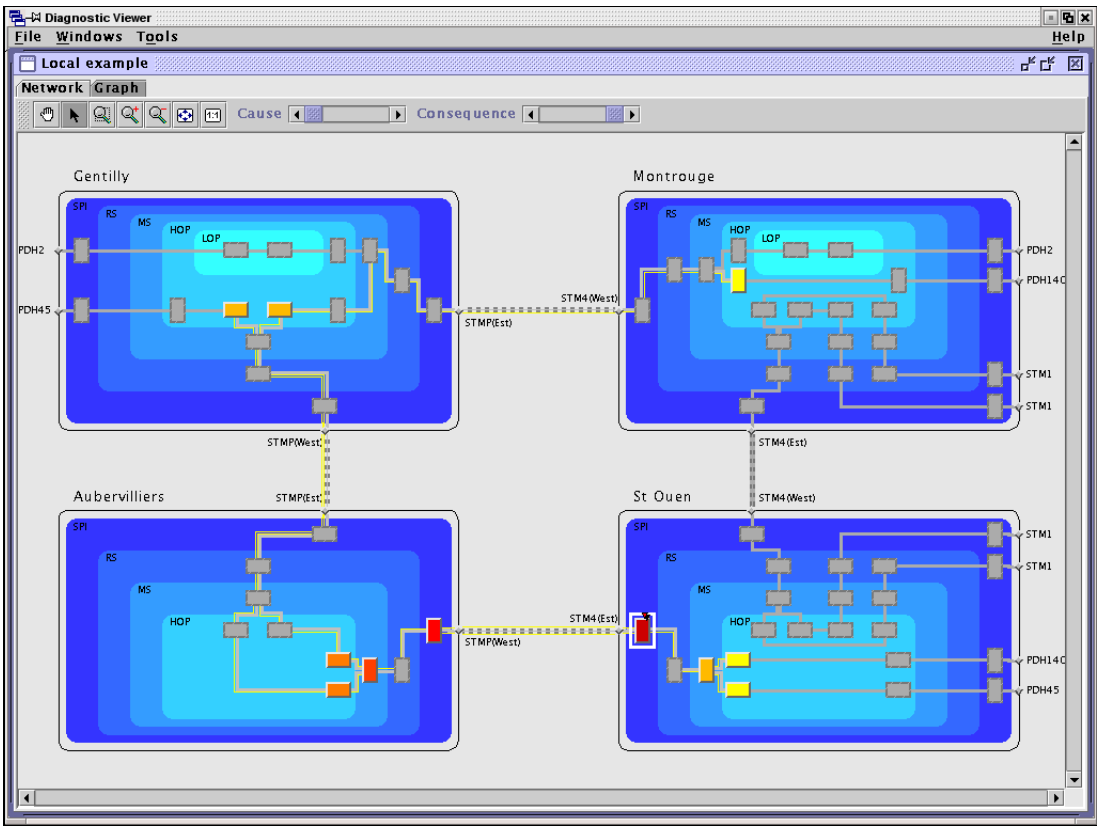

FIG. 6.12 – Interface graphique d'exploitation.

partir du résultat produit par Ddyp. Cette interface graphique est complémentaire de celle de Ddyp (voir figure 6.3). En effet, l'interface propre à Ddyp est générique, elle ne dépend pas du réseau supervisé. Elle permet de parcourir l'ensemble des hypothèses de diagnostic, de plus elle permet à l'interface d'Ilog d'afficher plusieurs hypothèses à la demande.

6.4.3.5 Résultats

Cette expérimentation a consisté à simuler des scénarios de pannes pré-établis à l'aide d'un simulateur de réseau. L'objectif de la chaîne a été de récupérer en ligne les alarmes produites par le simulateur afin d'établir le diagnostic et de l'afficher en ligne à l'aide de l'interface graphique (voir figure 6.12). Chaque scénario testé présente l'occurrence d'une ou de deux pannes primaires ayant lieu sur le réseau. La réponse du réseau à ces pannes est constituée d'un vingtaine d'alarmes au plus. Du fait de la nature du réseau, ces alarmes sont produites en cascade, nous avons donc considéré que l'ensemble d'alarmes produit par chaque scénario faisait partie d'une seule fenêtre temporelle.

Contrairement au réseau Transpac, l'anneau SDH est moins sujet au phénomène de masquage. Ayant un ensemble d'observations données, les comportements diagnostiqués sont moins nombreux. Une conséquence directe de cette propriété est que Ddyp est plus efficace sur l'anneau SDH que sur des réseaux du type Transpac. Dans la chaîne de diagnostic, les temps de réponses de Ddyp face aux scénarios testés sont corrects (moins de 10 secondes), ce qui permet aux opérateurs de pouvoir exploiter le diagnostic établi rapidement.

6.5 Conclusion

Ce chapitre a présenté un logiciel pour le diagnostic décentralisé de systèmes dynamiques tels que les réseaux de télécommunications : Ddyp. Il s'agit d'une application distribuée implantant les différents aspects de l'approche décentralisée (diagnostiqueur local, fusion, calcul de la stratégie de fusion...). Pour utiliser Ddyp, il suffit de définir un modèle du système à superviser et de choisir un déploiement des différents modules (objets Corba) de l'application adapté au système supervisé.

Cette application nous a permis de valider l'approche décentralisée sur différents systèmes issus de cas réels. Au niveau du diagnostic local, son calcul est très efficace grâce à l'utilisation des structures de diagnostiqueur (voir section 4.3.4.2). Ces études expérimentales montrent aussi l'intérêt du calcul de la stratégie de fusion dans une telle approche. Sans une telle stratégie, le calcul du diagnostic du système serait très inefficace et donc son exploitation en ligne irréaliste.

Ddyp a également été utilisé dans la mise en place d'une *chaîne de diagnostic* allant du gestionnaire chargé de récupérer les alarmes du système jusqu'à l'interface graphique proche des interfaces classiques de supervision permettant d'afficher le diagnostic des alarmes reçues. Cette réalisation a été possible du fait que Ddyp peut communiquer avec d'autres modules (gestionnaires, interfaces graphiques, ...) à l'aide d'un bus Corba.

CONCLUSION

L'objectif de cette thèse a été la mise au point d'une approche décentralisée pour le diagnostic de systèmes dynamiques tels que les réseaux de télécommunications. Dans un premier temps, nous avons établi que le diagnostic d'un tel système exigeait non seulement de rendre compte du dysfonctionnement de tel ou tel composant, mais qu'il fallait de plus pouvoir être en mesure de présenter à l'opérateur des explications complètes des observations reçues : les propagations de pannes. Les techniques existantes ne permettent pas ce genre de résultat car elles s'appliquent sur des systèmes dont la taille est raisonnable pour adopter une technique à base de modèle telle que l'approche diagnostiqueur de [Sampath et al. 95] ou bien parce qu'elles donnent une information moins riche telle que la détection voire l'identification de la panne mais pas une explication complète de ce qui a pu se passer.

L'approche décentralisée que nous avons développée au cours de cette thèse est bien adaptée pour deux raisons.

1. Les systèmes étudiés sont répartis, les observations sont issues de sites différents, ce qui induit des problèmes liés à l'observabilité du système. Plus les propriétés sur l'observabilité du système sont strictes (connaissances importantes de relation de précédence temporelle entre les observations) plus il est aisé de proposer un diagnostic exhaustif du système. Une architecture de type décentralisé aide à cela en permettant de délocaliser les diagnostiqueurs aux endroits les plus « pertinents » pour l'observation de tel ou tel site.
2. Les systèmes étudiés sont de grande taille. La quantité d'informations à traiter est importante (taille du système, nombre de composants élémentaires, nombre d'alarmes reçues...). Appliquer des techniques de diagnostic centralisées est impossible si l'on cherche à donner une interprétation des alarmes fines telle que la propagation des pannes expliquant ce flot. Là encore, l'approche décentralisée apporte sa contribution : l'information observée est traitée en deux phases, la première consistant à établir des diagnostics locaux en fonction des observations locales et la deuxième consistant à fusionner ces diagnostics locaux en vue d'établir le diagnostic du système.

Afin de rendre cette approche opérationnelle, nous avons proposé des algorithmes pour résoudre le problème du diagnostic le plus efficacement possible, en « cassant » la complexité du problème quant cela était possible.

Recenser l'ensemble des séquences d'événements ayant pu se produire sur le système et expliquant un ensemble d'observations données peut être complexe en temps et en espace. Cette complexité vient essentiellement de la nature répartie du système qui a la particularité de produire des événements concurrents indépendants. Afin de résoudre ce problème, nous proposons de recenser les hypothèses de diagnostic comme un ensemble de traces d'événements, chaque trace représentant un ensemble d'hypothèses de diagnostic à la concurrence d'événements près.

Le deuxième point d'optimisation porte sur le calcul du diagnostic local. L'algorithme de base est une recherche de comportements locaux fondés sur les observations locales. Cette re-

cherche peut être coûteuse dès lors que les composants diagnostiqués ont des comportements non observables importants. Nous avons donc mis en place une structure de données augmentant l'efficacité du calcul du diagnostic local : cette structure appelée *diagnostiqueur* est une extension de celle proposée par [Sampath et al. 95] proposant un diagnostic enrichi avec les interactions éventuelles avec le voisinage.

Une autre difficulté a concerné la fusion des diagnostics. Cette fusion est une opération nécessaire afin de valider les hypothèses locales : cette opération peut être coûteuse si elle est appliquée de manière intempestive. Nous proposons donc d'appliquer des fusions que lorsque cela est nécessaire. Cette nécessité est détectée par l'application d'*une stratégie de fusion* calculée dynamiquement en fonction des diagnostics locaux courants. Cette stratégie est établie en fonction des interactions proposées par les différents diagnostics locaux. Le résultat d'une telle stratégie est que le diagnostic global est représenté par un ensemble de *diagnostics indépendants* qui représentent les propagations des pannes du système qui ont pu avoir lieu en concurrence à un instant donné.

La dernière difficulté à laquelle il a fallu faire face est la quantité d'observations à traiter et le caractère en ligne du diagnostic. Nous proposons de découper le temps en *fenêtres temporelles* et d'y appliquer les algorithmes précédemment cités pour chaque fenêtre temporelle. Ici la difficulté est liée à l'observabilité du système. Peut-on être sûr que les observations reçues jusqu'à maintenant peuvent me permettre d'établir un diagnostic ou en manque-t-il ? Suivant la réponse à cette question, le traitement incrémental du diagnostic est différent, soit les fenêtres temporelles sont *sûres* et il est facile de calculer le diagnostic d'une nouvelle fenêtre temporelle en fonction de la précédente, soit elles ne le sont pas, dans ce cas, nous proposons d'anticiper l'apparition d'observations manquantes afin d'assurer le fait qu'il est toujours possible d'établir le diagnostic d'une nouvelle fenêtre temporelle en fonction de la précédente.

Le résultat de cette thèse a été le développement et la mise au point d'une plate-forme de diagnostic mettant en œuvre tous les principes développés : la plate-forme Ddyp. Cette plate-forme est facilement adaptable à tout type de systèmes dynamiques à événements discrets car son unique point d'entrée est le modèle associé à ce système. Dans le cadre de Magda, nous avons déployé cette plate-forme afin de la connecter à un gestionnaire d'alarmes recevant les alarmes du réseau et à une interface graphique dédiée directement exploitable par un opérateur de supervision : cet ensemble constitue la *chaîne de supervision* complète entre le réseau et son superviseur.

Perspectives

Les perspectives liées à ce travail sont nombreuses. Quatre axes complémentaires de recherche peuvent être dégagés.

Robustesse de l'approche

Dans le cadre de cette thèse, nous avons toujours considéré que le modèle du système était connu *a priori* et qu'il était supposé complet. Cette hypothèse doit être levée afin de gérer le fait qu'on peut ne pas être en mesure de construire un modèle complet. Une conséquence de l'incomplétude du modèle est le fait qu'il n'est parfois plus possible d'établir un diagnostic (par

exemple, la signature observée n'appartient pas au comportement observable du système). On peut voir deux axes possibles à poursuivre. Le premier consiste à mettre au point un système de suivi robuste en utilisant un modèle d'incomplétude (modèle de perte d'alarmes probabiliste par exemple, ou graphe d'observations incertaines [Lamperti et Zanella 00]) qui permettrait la possibilité de reprendre le diagnostic « dès que l'on reconnaît à nouveau une signature d'observations ». L'avantage de cette approche serait son efficacité en ligne mais son inconvénient majeur est qu'elle n'est pas en mesure de tirer les leçons des situations passées. Le deuxième axe consiste plutôt à mettre en place un système qui soit en mesure d'apprendre les comportements inconnus. On pourrait imaginer que ce système propose un ensemble d'hypothèses à l'opérateur dont la tâche serait de les valider ou de les invalider. En fonction des réponses de l'opérateur, le système adapte le modèle en ligne.

Vers la gestion des reconfigurations

Le système peut être sujet à des *reconfigurations* dynamiques (modifications de la topologie des connexions virtuelles dans un réseau par exemple) qu'il serait intéressant de suivre au même titre que les observations. En effet, une reconfiguration produit un ensemble de nouveaux comportements et le système de supervision doit être en mesure de suivre ces reconfigurations afin d'expliquer des pannes en rapport avec l'occurrence de reconfigurations dynamiques. Afin de gérer ces reconfigurations, l'idée serait de mettre en place un *modèle de reconfiguration* qui prenne en compte les notifications de reconfigurations. Ce modèle serait utilisé en ligne afin d'avertir le système de diagnostic qu'une reconfiguration a eu lieu et que le modèle de diagnostic doit être adapté en conséquence (cette adaptation peut être connue à partir des MIB par exemple) à l'aide d'une bibliothèque de modèles locaux qui peuvent être dynamiquement chargés.

Vers l'autonomie du système de supervision

Dans une optique à plus long terme, un axe de recherche intéressant serait d'étudier les liens entre le diagnostic décentralisé et la planification distribuée de systèmes autonomes afin de développer des méthodes de reconfiguration automatique. Le diagnostic permettrait ainsi au système d'appliquer un plan de reconfiguration afin de réparer le dysfonctionnement diagnostiqué. L'intégration de techniques de diagnostic avec des techniques de planification permet de donner à de tels systèmes la possibilité de se reconfigurer automatiquement (auto-réparation), ce qui augmente ainsi leur fiabilité et leur autonomie.

Cet axe de recherche n'est pas uniquemement lié à la gestion des réseaux de télécommunications mais aussi à d'autres systèmes complexes et autonomes [Williams et Nayak 96], en particulier les systèmes de production et de distribution d'énergie [Thiébaux et al. 94], de contrôle de processus chimiques, les systèmes de satellites...

Utilisation des outils de vérification de modèles

Le diagnostic d'un système dynamique représenté par un modèle à événements discrets est fondé sur la recherche de chemins, de séquences d'événements et d'états du système ; il

peut ainsi être vu comme la solution à un problème d'atteignabilité. Dans le domaine de la vérification de modèles, ce type de problème a été étudié et a abouti aux développements d'outils puissants capables de déduire d'un modèle ce type de propriété de façon la plus efficace possible. Un dernier axe de recherche serait donc d'étudier les moyens pour traduire le problème du diagnostic afin de mettre à profit la puissance des outils de vérification de modèles [Cordier et Largouët 01].

ANNEXE *A*

Modèle de Toynet

Nous présentons dans cet annexe le modèle complet du réseau Toynet.

Modèle comportemental

Ce modèle est constitué de 12 composants élémentaires. Il existe 4 types de composants élémentaires. Nous présentons sur les figures A.1 A.2 A.3 et A.4 un composant élémentaire de chaque type.

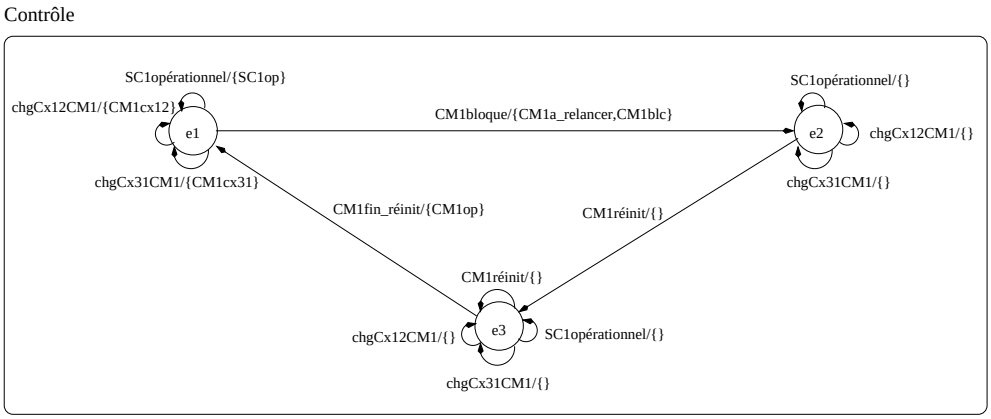

FIG. A.1 – Composant élémentaire représentant la partie *contrôle* de l'équipement *CM1*.

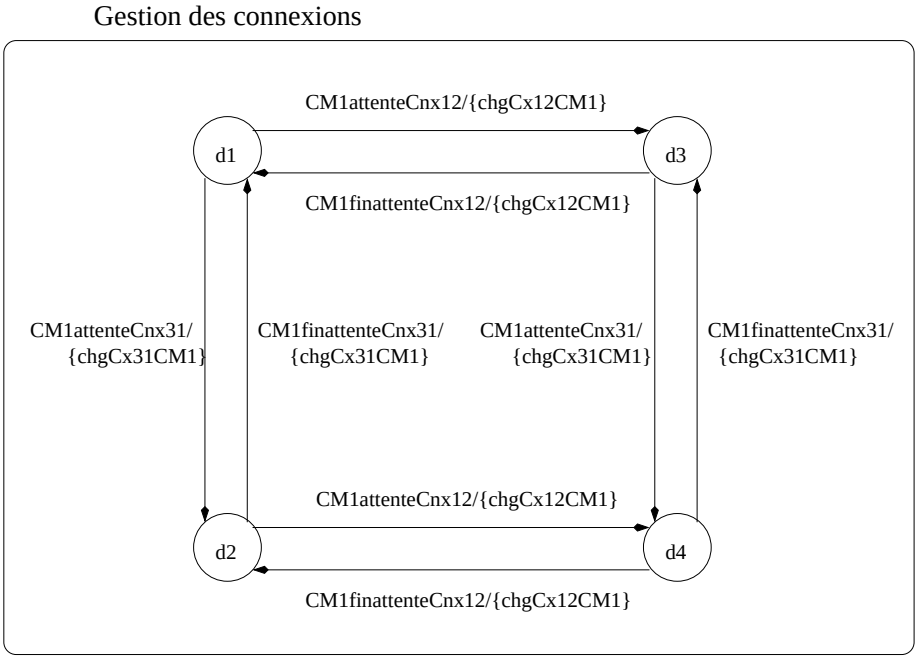

FIG. A.2 – Composant élémentaire représentant la partie *gestion des connexions* de l'équipement *CM1*.

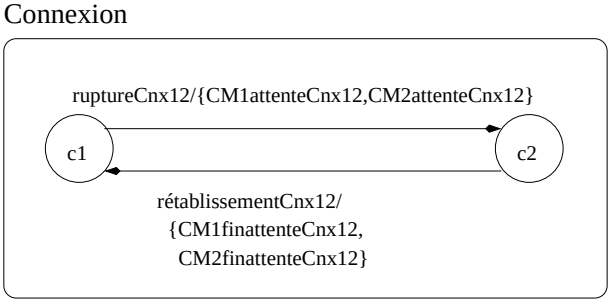

FIG. A.3 – Composant élémentaire représentant la connexion *Cnx12*.

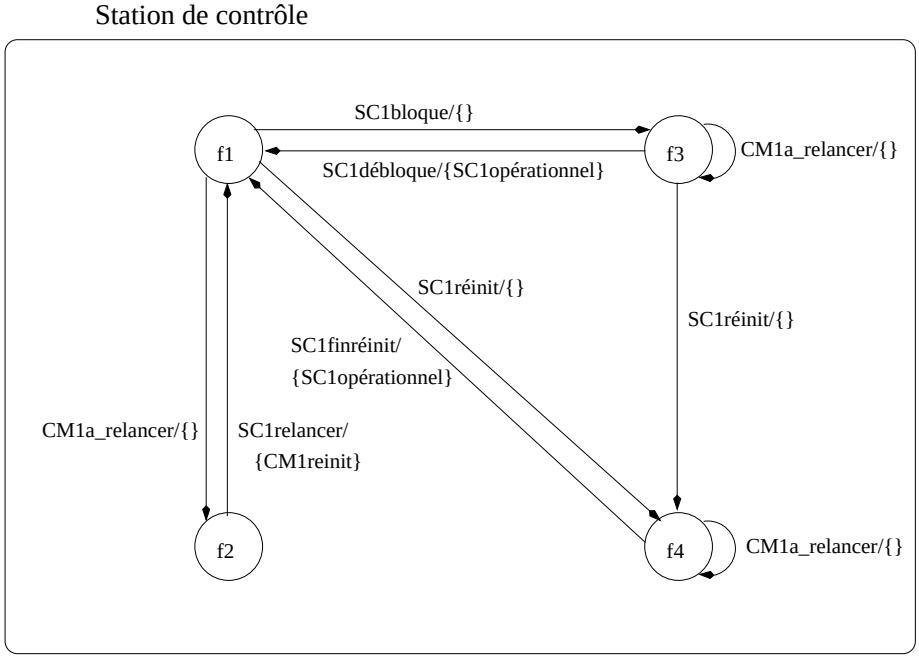

FIG. A.4 – Composant élémentaire représentant la station de contrôle *SC1*.

Modèle structurel

La figure A.5 présente le modèle structurel de Toynet.

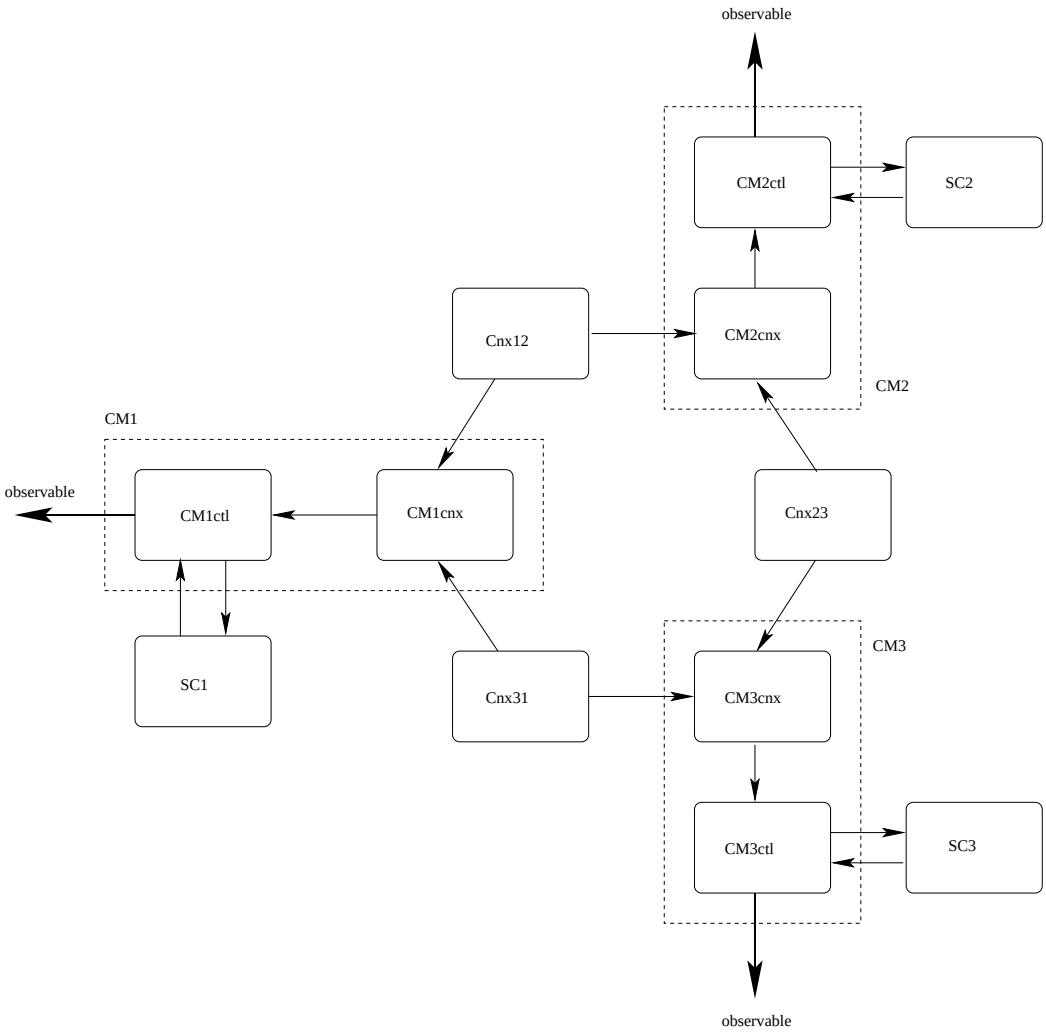

FIG. A.5 – Modèle structurel de Toynet.

ANNEXE *B*

Spécification du langage de description des modèles

Nous présentons dans cette annexe les règles du langage servant à décrire les modèles de systèmes à superviser. Ce langage est LL(1).

```
specification:
        SPECIFICATION IDENTIFIER SEMICOLON
        first_module_definition
        END POINT


first_module_definition:
        module_header_definition
        first_module_body_definition


first_module_body_definition:
        BEHAVIOR body_identifier FOR header_identifier SEMICOLON
        first_body_definition

first_body_definition
        module_definition
        module_definition_list
        END SEMICOLON
        structural_initialization

/***************** Module definition ******************/

module_definition:
        module_header_definition
        module_body_definition

module_definition_list:
        | module_definition module_definition_list

/***************** Module header declaration **************/
```

```
module_header_definition:
        MODULE header_identifier SEMICOLON
        module_header_definition_continued


module_header_definition_continued:
          interaction_point_declaration_part END SEMICOLON


header_identifier:
        IDENTIFIER

/****************** Interface point declaration ***********/

interaction_point_declaration_part:
        IP interaction_point_declaration_rec

interaction_point_declaration_rec:
          interaction_point_declaration SEMICOLON
        | interaction_point_declaration SEMICOLON
          interaction_point_declaration_rec

interaction_point_declaration:
        interaction_point_role interaction_point_declaration_continued

interaction_point_declaration_continued:
          IDENTIFIER
        | IDENTIFIER ip_identifier_list COLON interaction_list

interaction_point_role:
          INPUT
        | OUTPUT
        | OBSERVABLE

ip_identifier_list:
        | ip_identifier ip_identifier_list


ip_identifier:
        IDENTIFIER


interaction_list:
        LBRACKET interaction_definition_list RBRACKET

interaction_definition_list:
        interaction_definition
        | interaction_definition COMMA interaction_definition_list
```

```
interaction_definition:
        interaction_identifier


interaction_identifier:
          IDENTIFIER
        | IDENTIFIER LBRACKET IDENTIFIER RBRACKET


/***************** Module body declaration ****************/

module_body_definition:
        BEHAVIOR body_identifier FOR header_identifier SEMICOLON
        body_definition


body_identifier:
        IDENTIFIER

body_definition:
          elementary_module END SEMICOLON
        | module_definition
          module_definition_list
          END SEMICOLON
          structural_initialization

elementary_module:
        state_definition_part
        initialization_part
        transition_declaration_part

/***************** State declaration ***********************/

state_definition_part:
        STATE state_identifier_list SEMICOLON

state_identifier_list:
          state_identifier
        | state_identifier COMMA state_identifier_list

state_identifier:
        IDENTIFIER

/***************** Transition definition ************************/

initialization_part:
        INITIALIZE to_clause SEMICOLON

transition_declaration_part:
```

```
        | transition_declaration
        transition_declaration_part

transition_declaration:
        TRANS transition_group

transition_group:
          clause_group SEMICOLON

clause_group:
        from_clause
        to_clause
        when_clause
        output_clause


from_clause:
        FROM state_identifier_list


to_clause:
        TO to_state_identifier


to_state_identifier:
        state_identifier

when_clause:
        WHEN interaction_reference


output_clause:
        | OUTPUT interaction_reference
          output_clause

interaction_reference:
        IDENTIFIER POINT interaction_identifier


/**************** Structural Initialization *************/

structural_initialization:
        STRUCTURE
        attach_statement_list
        connect_statement_list
        END SEMICOLON


attach_statement_list:
```

```
        | attach_statement attach_statement_list


attach_statement:
        ATTACH external_ip TO child_ip SEMICOLON

child_ip:
        module_identifier POINT external_ip

module_identifier:
        IDENTIFIER

external_ip:
        IDENTIFIER


connect_statement_list:
        | connect_statement connect_statement_list


connect_statement:
        CONNECT output_child_ip TO input_child_ip SEMICOLON

output_child_ip:
        child_ip

input_child_ip:
        child_ip
```

ANNEXE *C*

Spécification technique de la plate-forme Ddyp

Le développement de Ddyp a abouti à la réalisation de plusieurs bibliothèques C++, chaque bibliothèque mettant en œuvre une des fonctionalités de Ddyp.

1. `libGraph.so` : bibliothèque de base sur les graphes.
2. `libModel.so` : bibliothèque mettant en œuvre le modèle.
3. `libLocalDiag.so` : bibliothèque du diagnostic local.
4. `libDiagnosis.so` : bibliothèque de diagnostic.

Nous présentons dans cet annexe une vue simplifiée de l'architecture des deux bibliothèques principales : `libModel.so` et `libDiagnosis.so`.

C.1 Bibliothèque - Acquisition des modèles

La bibliothèque `libModel.so` met en œuvre tous les outils nécessaires à l'exploitation d'un modèle décrit dans un fichier avec le langage spécifié dans l'annexe B. Cette bibliothèque dispose des fonctionnalités suivantes :

1. acquisition d'un modèle à partir d'un fichier de description (voir annexe B) ;
2. vérification de la cohérence des connexions entre les différents composants ;
3. composition de modèles locaux en vue d'obtenir un modèle plus global.

L'architecture de cette bibliothèque est présentée par le diagramme de classe UML de la figure C.1. Un modèle est représenté par une instance de la classe *Module* (voir section 6.2.1.2 page 141). Une instance de la classe *Module* met en œuvre un *module non-élémentaire* : elle contient un ensemble de modules fils (élémentaires (*ElementaryModule*) ou non (*Module*)), des connexions (*Connection*) et des attachements (*AttachInPort*, *AttachOutPort*) entre des ports (ports d'entrée (*InPort*) et port de sortie (*OutPort*)). Un événement est représenté par l'occurrence d'un message (*Message*) sur un port (*Port*). Si le port est un port d'entrée (*InPort*), l'événement est un événement de réception (*InEvnt*), si le port est un port de sortie (*OutPort*), l'événement est un événement d'émission (*OutEvnt*). Le comportement d'un composant élémentaire est décrit dans une instance de la classe *ElementaryModule* : cette classe met en œuvre les transducteurs constitués d'états (*StateLabel*) et de transitions (*TransLabel*).

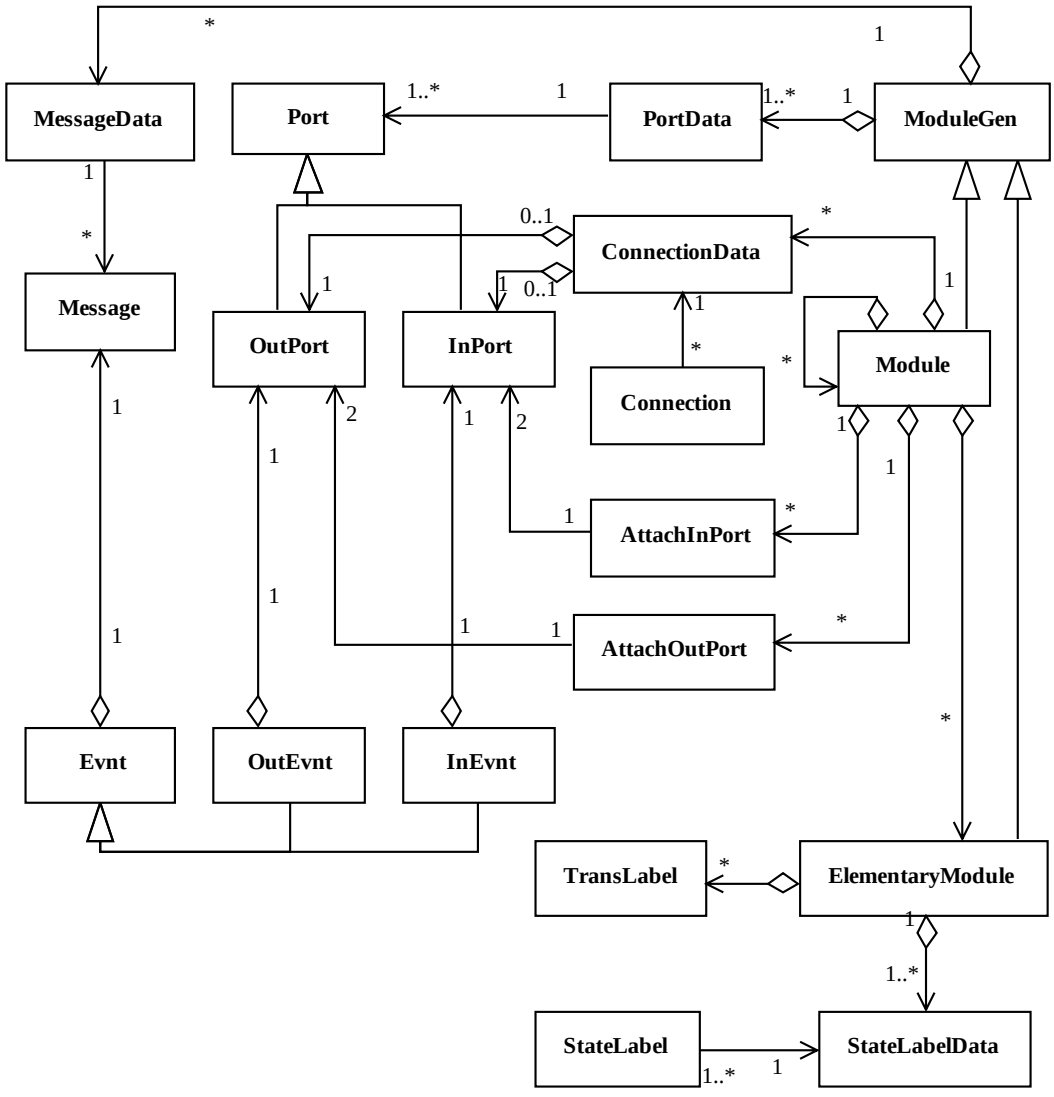

FIG. C.1 – Diagramme simplifié des classes de la bibliothèque *Model*.

C.2 Bibliothèque - Calcul des diagnostics

Cette bibliothèque met en œuvre les structures de données nécessaires au calcul du diagnostic. Cette bibliothèque est fondée sur la notion d'identificateurs (*Identifier*) et propose une vue abstraite du modèle décrit par la bibliothèque *Model*. Tout module élémentaire est représenté par un *Component*. Tout événement (*InEvnt* ou *OutEvnt*) est associé à un identifiant *Event*, quant aux alarmes (ce sont des événements de sortie particuliers), elles sont asssociées à des identifiants de la classe *Observation*... La structure de donnée de base est *DiagnosisGen* qui met en œuvre la représentation basique du diagnostic (système de transitions (*StateDiagnosis*, *TransDiagnosis*)). Un objet de type *StateDiagnosis* est associé à un ensemble d'états du modèle (représentés par des objets de type *StateId*) et un ordre partiel d'observations *PartialOrderSet*. Quant à un objet de type *TransDiagnosis*, il est associé à une transition du modèle (représentée par un objet de type *TransId*). La classe *DiagnosisGen* est dérivée en deux sous-classes : la classe *Diagnosis* met en œuvre la représentation non-réduite du diagnostic (voir section 4.2.1 page 81) et la classe *ReducedDiagnosis* met en œuvre la représentation réduite (voir section 4.2.4 page 89).

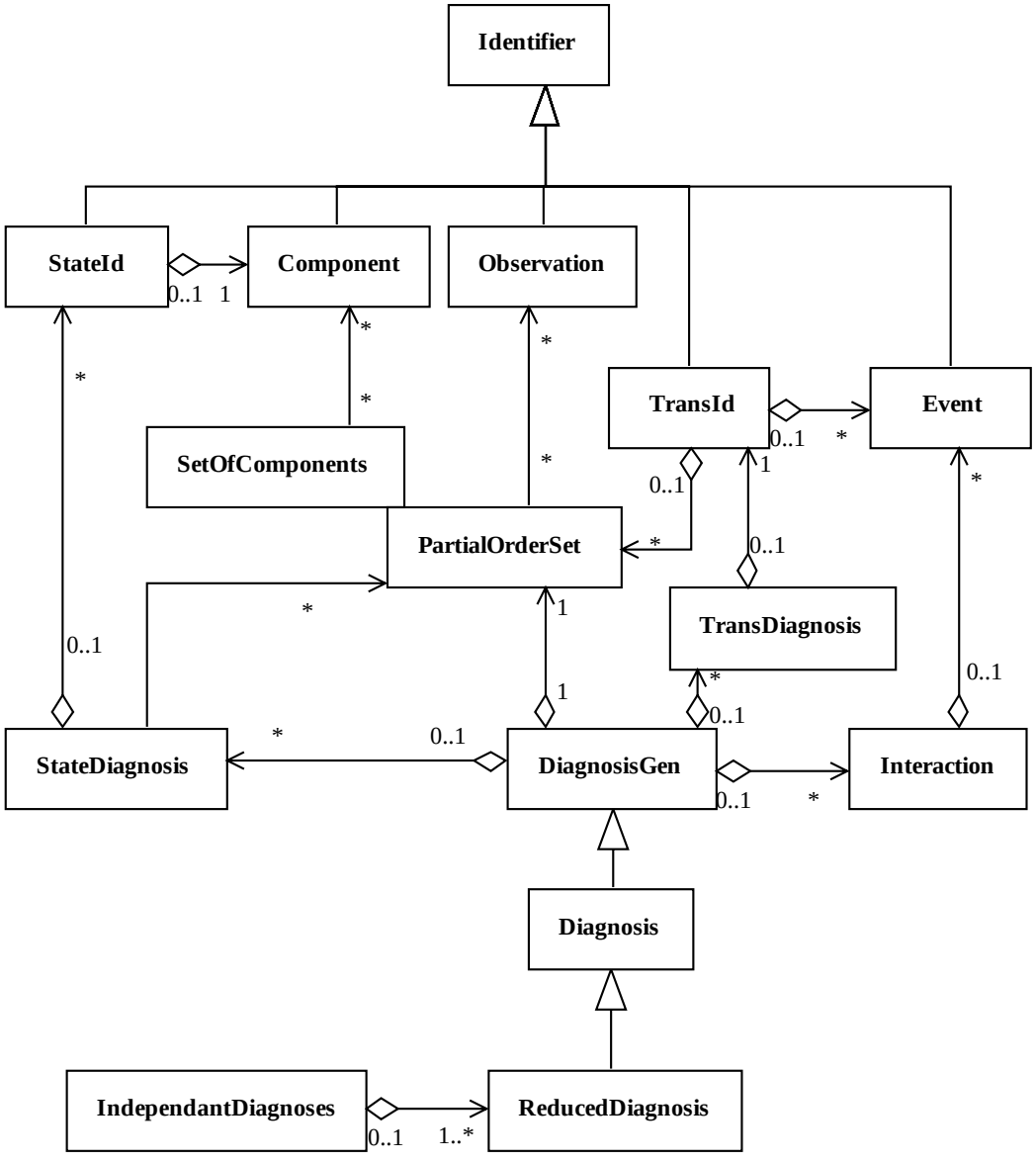

FIG. C.2 – Diagramme simplifié des classes de la bibliothèque *Diagnosis*.

BIBLIOGRAPHIE

[Aghasaryan et al. 98] A. AGHASARYAN, E. FABRE, A. BENVENISTE, R. BOUBOUR et C. JARD, « Fault detection and diagnosis in distributed systems : an approach by partially stochastic Petri nets », *Discrete Event Dynamic Systems*, vol. 8, n° 2, juin 1998, p. 203–231, Special issue on Hybrid Systems.

[Aghasaryan 98] A. AGHASARYAN, *Formalisme HMM pour les réseaux de Petri partiellement stochastiques : Application au diagnostic de pannes dans les systèmes répartis*, Irisa, Campus de Beaulieu, F-35042 Rennes Cedex, thèse, SPM/Université de Rennes 1, décembre 1998.

[Aho et Ullman 72] A.V. AHO et J.D. ULLMAN, *The Theory of Parsing, Translation, and Compiling*, Prentice-Hall, 1972, *series in automatic computation*.

[Arnold et Nivat 82] A. ARNOLD et M. NIVAT, « Comportement de processus », *Colloque* AFCET *"Les Mathématiques de l'Informatique"*, p. 35–68, 1982.

[Arnold 92] A. ARNOLD, *Systèmes de transitions finis et sémantique des processus communicants*, Masson, 1992.

[Arpège 92] ARPÈGE, *Gestion de réseaux : concepts et outils*, Masson, 1992.

[Azarmi et al. 93] N. AZARMI, S. AZMOODEH, J. BIGHAM et D. PANG, « Model-Based Diagnosis in Maintenance of Telecommunication Networks », *Proceedings of the International Workshop on Principles of Diagnosis*, p. 46–59, 1993.

[Baroni et al. 98] P. BARONI, G. LAMPERTI, P. POGLIANO et M. ZANELLA, « Diagnosis of active systems », *Proceedings of the European Conference on Artificial Intelligence (ECAI'98)*, Henry Prade ed., John Wiley & Sons, 1998.

[Baroni et al. 99] P. BARONI, G. LAMPERTI, P. POGLIANO et M. ZANELLA, « Diagnosis of large active systems », *Artificial Intelligence*, vol. 110, 1999, p. 135–183.

[Baroni et al. 00] P. BARONI, G. LAMPERTI, P. POGLIANO et M. ZANELLA, « Diagnosis of a Class of Distributed Discrete-Event Systems », *IEEE Transactions on systems, man, and cybernetics*, vol. 30, n° 6, novembre 2000, p. 731–752.

[Basseville et Nikiforov 93] M. BASSEVILLE et I. NIKIFOROV, *Detection of abrupt changes - Theory and applications*, Prentice-Hall, 1993.

[Berstel 02] B. BERSTEL, « Intégration de la reconnaissance de chroniques à un algorithme incrémental de pattern matching », *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'2002)*, 2002.

[Bibas et al. 96] S. BIBAS, M.-O. CORDIER, P. DAGUE, F. LÉVY et L. ROZÉ, « GASPAR: a model-based system for diagnosing telecommunication networks », *IMACS-IEEE/SMC International Multiconference of Computational Engineering in Systems Applications (CESA'96)*, Lille, 1996.

[Bigham et al. 92] J. BIGHAM, D. PANG et T. CHAU, « A generic maintenance for telecommunication networks », *Proceedings of the conference on management of telecommunication networks*, E. Horwood ed., p. 195–211, 1992.

[Boubour 97] R. BOUBOUR, *Suivi de pannes par corrélation causale d'alarmes dans les systèmes répartis : Application aux réseaux de télécommunication*, Irisa, Campus de Beaulieu, F-35042 Rennes Cedex, thèse, Ifsic/Université de Rennes 1, 1997.

[Bouloutas et al. 92] A.T. BOULOUTAS, G.W. HART et M. SCHWARTZ, « Simple finite-state fault detectors for communication networks », *IEEE Transactions on Communications*, vol. 40, n° 3, 1992.

[Bouloutas et al. 94] A.T. BOULOUTAS, S.B. CALO et A. FINKEL, « Alarm correlation and fault identification in communication networks », *IEEE Transactions on Communications*, vol. 42, n° 2/3/4, 1994.

[Bouyer 97] G. BOUYER, *Les réseaux synchrones étendus PDH et SDH*, Hermès, 1997.

[Bratko et al. 88] I. BRATKO, I. MOZETIC et N. LAVRAC, « Automatic synthesis and compression of cardiological knowledge », *Proceedings of Machine Intelligence*, E. Horwood ed., p. 435–454, 1988.

[Cassandras et Lafortune 99] C.G. CASSANDRAS et S. LAFORTUNE, *Introduction to discrete event systems*, Kluwer Academic Publishers, 1999.

[Cauvin et al. 92] S. CAUVIN, B. BRAUNSCHWEIG, P. GALTIER et Y. GLAIZE, « Alexip, expert system coupled with a dynamic simulator for the supervision of the alphabutol process », *Revue of Institut Français du Pétrole*, vol. 47, 1992, p. 375–382.

[Chirashnya et al. 01a] I. CHIRASHNYA, A. IVTSAN, L. SHALEV et K. SHOIKET, « Combining Reliability Theory and Model-Based Diagnosis For Switched Networks Diagnostics », *Proceedings of the International Workshop on Principles of Diagnosis (DX'01)*, p. 23–30, Sansicario, Itale, mars 2001.

[Chirashnya et al. 01b] I. CHIRASHNYA, A. IVTSAN, L. SHALEV et K. SHOIKET, « Modeling Complex, Heterogeneous and Dynamic Networking Environments for Diagnostics », *Proceedings of the International Workshop on Principles of Diagnosis (DX'01)*, p. 31–38, Sansicario, Italie, mars 2001.

[Console et Torasso 92] L. CONSOLE et P. TORASSO, « A spectrum of logical definitions of model-based diagnosis », *Computational Intelligence*, vol. 7, n° 3, 1991 (repris dans *Readings in Model-Based Diagnosis*, W. Hamscher, L. Console, J. de Kleer (dir.), Morgan Kaufmann,p. 78–88, 1992), p. 133–141.

[Cordier et Largouët 01] M-O. CORDIER et C. LARGOUËT, « Using model-checking techniques for diagnosing discrete-event systems », *Proceedings of the International Workshop on Principles of Diagnosis (DX-01)*, p. 39–46, Sansicario, Italie, 2001.

[Debouk et al. 98] R. DEBOUK, S. LAFORTUNE et D. TENEKETZIS, « A coordinated decentralized protocol for failure diagnosis of discrete event systems », *Proceedings of the Workshop on Discrete Event Systems (WODES'98)*, p. 138–143, Cagliari, Italie, 1998.

[Debouk et al. 00a] R. DEBOUK, S. LAFORTUNE et D. TENEKETZIS, « Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems », *Discrete Event Dynamic Systems*, vol. 10, n° 1-2, 2000, p. 33–86.

[Debouk et al. 00b] R. DEBOUK, S. LAFORTUNE et D. TENEKETZIS, « On the Effect of Communication Delays in Failure Diagnosis of Decentralized Discrete Event Systems », *Proceedings of IEEE Conference on Decision and Control 2000*, Sydney, Australie, 2000.

[Didelet et Dubuisson 92] E. DIDELET et B. DUBUISSON, « A diagnostic system for the French long distance network using neural trees and a rule-based system », *Proceedings of International Conference on Systems, Man and Cybernetics*, p. 717–722, Chicago, Etats-Unis, 1992.

[Didelet 92] E. DIDELET, *Les arbres de neurones avec rejet d'ambiguïté. Application au diagnostic pour le pilotage en temps réel du réseau téléphonique français*, thèse, Université de technologie de Compiègne, 1992.

[dK et Williams 92] J. DE KLEER et B. C. WILLIAMS, « Diagnosis with behavioral modes », *Proceedings of the 11th International Joint Conference on Artificial Intelligence IJCAI'89*, p. 1324–1330, Detroit, MI, Etats-Unis, 1989 (repris dans *Readings in Model-Based Diagnosis*, W. Hamscher, L. Console, J. de Kleer (dir.), Morgan Kaufmann,p. 124–130, 1992).

[dKleer et al. 92] J. DE KLEER, A. MACKWORTH et R. REITER, « Characterizing diagnoses and systems », *Artificial Intelligence*, vol. 56, n° 2–3, 1992 (repris dans *Readings in Model-Based Diagnosis*, W. Hamscher, L. Console, J. de Kleer (dir.), Morgan Kaufmann,p. 54–65, 1992), p. 197–222.

[Dousson et al. 93] C. DOUSSON, P. GABORIT et M. GHALLAB, « Situation recognition: representation and algorithms », *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, p. 166–172, Chambéry, France, 1993.

[Dousson et Du'o'ng 99] C. DOUSSON et T. VU DU'O'NG, « Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems », *Proceedings of the 16th International Joint Conference on Artificial Intelligence IJCAI'99*, 1999.

[Du'o'ng 01] T. VU DU'O'NG, *Découverte de chroniques à partir de journaux d'alarmes Application à la supervision de réseaux de télécommunications*, France Télécom R&D, Lannion, thèse, Institut national polytechnique de Toulouse, 2001.

[Fabre et al. 00] E. FABRE, A. BENVENISTE, C. JARD, L. RICKER et M. SMITH, « Distributed State Reconstruction for Discrete Event Systems. », *Proc. of the 2000 IEEE Control and Decision Conference (CDC)*, Sydney, Australie, 2000.

[Fidge 88] J. FIDGE, « Time stamps in Message Passing Systems that Preserve the Partial Ordering », *Proceedings of 11th Australian Computer Science Conference*, p. 55–66, 1988.

[Forney 73] G.D. FORNEY, « The Viterbi Algorithm », *Proceedings of the IEEE*, vol. 61, n° 3, 1973.

[Frydman et al. 01] C. FRYDMAN, M. LE GOC, L. TORRES et N. GIAMBIASI, « The diagnosis approach used in SACHEM », *Working notes of the 12th International Workshop on Principles of Diagnosis (DX'01)*, p. 63–70, Sansicario, Italie, 2001.

[Godefroid et Wolper 91] P. GODEFROID et P. WOLPER, « Using Partial Orders for the Efficient Verification of Deadlock Freedom and Safety Properties », *Proceedings of the 3rd International Conference on Computer Aided Verification (CAV'91)*, p. 332–342, Springer-Verlag, Aalborg, Danemark, 1991.

[Goodman et al. 95] R.M. GOODMAN, B.E. AMBROSE, H. W. LATIN et C. T. ULMER, « NOAA - An expert System managing the Telephone Network », *Proceedings of the conference on Integrated Network Management*, A.S. Selthi, Y. Raynaud et F. Faure-Vincent ed., p. 316–327, Chapman and Hall, 1995.

[Group 01] OBJECT MANAGEMENT GROUP, *The Common Object Request Broker: Architecture and Specification (Revision 2.6)*, 2001.

[Gruschke 98a] B. GRUSCHKE, « Integrated event management: event correlation using dependency graphs », *Proceedings of DSOM'98*, 1998.

[Gruschke 98b] B. GRUSCHKE, « A New Aproach for Event Correlation based on Dependency Graphs », *Proceedings of the 5th workshop of the OpenView University Association: OVUA'98*, Rennes, France, 1998.

[Gurer et al. 95] D.W. GURER, I. KHAN, R. OGIER et R. KEFFER, « An artificial intelligence approach to network fault management », *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[Hamscher et al. 92] W. HAMSCHER, L. CONSOLE et J. DE KLEER (éd.), *Readings in Model-Based Diagnosis*, Morgan Kaufmann, San Meteo, CA, Etats-Unis, 1992.

[Hong et Sen 91] P. HONG et P. SEN, « Incorporating non-deterministic reasoning in managing heterogeneous network faults », *Integrated Network Management*, I. Krishnan et W. Zimmer ed., p. 481–492, Elsevier Science, 1991.

[ISO 97] ISO, *Information processing systems — Open systems interconnection — Estelle — A formal description technique based on an extended state transition model*, 1997.

[Jakobson et Weissman 93] G. JAKOBSON et D. WEISSMAN, « Alarm correlation », *IEEE network*, novembre 1993, p. 52–59.

[Katzela et al. 95] I. KATZELA, A.T. BOULOUTAS, S.B. CALO et A. FINKEL, « Schemes for Distributed Fault Identification in Communication Networks », *Journal of Network and System Management*, 1995.

[Kehl et al. 93] W. KEHL, F. MAIER, T. CHAU et G. SCHAPELER, « Model-based maintenance for telecommunication networks », *Proceedings of International conference of expert systems*, 1993.

[Kliger et al. 95] S. KLIGER, S. YEMINI, Y. YEMINI, D. OHSIO et S. STOLFO, « A Coding Approach to Event Correlation », *Integrated Network Management*, Sethi, Raynaud et Faure-Vincent ed., p. 266–277, IFIP, Chapman and Hall, mai 1995.

[Lamperti et Zanella 00] G. LAMPERTI et M. ZANELLA, « Uncertain Discrete-Event Observations », *Proceedings of the International Workshop on Principles of Diagnosis (DX'00)*, A. Darwiche et G. Provan ed., p. 101–108, 2000.

[Larsson 99] M. LARSSON, *Behavioral and Structural Model Based Approaches to Discrete Diagnosis*, thèse, Linköping Studies in Science and Technology. Thesis No 608, novembre 1999.

[Lewis 93] L. LEWIS, « A case-based reasoning approach to the resolution of faults in communication networks », *IFIP Transactions on Integrated Network Management III*, H.G. Hegering et Y. Yemeni ed., p. 671–682, Elsevier Science, 1993.

[MA 90] A. MOUNIR-ALAOUI, *Raisonnement temporel pour la planification et la reconnaissance de situations*, Toulouse, thèse, Université Paul Sabatier, 1990.

[Mattern 88] F. MATTERN, « Virtual Time and Global States of Distributed Systems », *Proceedings of International Workshop on Parallel and Distributed Algorithms*, octobre 1988.

[Mayer 99] E. MAYER, *Apprentissage inductif de scénarios pour la supervision de réseaux de télécommunications*, Irisa, Rennes, thèse, Université de Rennes 1, 1999.

[Mazurkiewicz 86] A. MAZURKIEWICZ, « Trace theory », *Petri Nets: Applications and relationships to Other Models of Concurrency, Advances in Petri Nets*, p. 279–324, 1986.

[Nygate 95] Y.A. NYGATE, « Event correlation using rule and object techniques », *Proceedings of the conference on Integrated Network Management*, A.S. Sethi, Y. Raynaud et F. Faure-Vincent ed., p. 251–261, Chapman and Hall, 1995.

[Osmani 99] A. OSMANI, *Diagnostic de pannes dans les réseaux : approche à base de modèles et raisonnment temporel*, LIPN, thèse, Université de Paris 13, 1999.

[Pearce 88] D.A PEARCE, « The induction of fault diagnosis systems from qualitative reasoning », *Proceedings of the AAAi National Conference on Artificial Intelligence*, p. 353–357, St Paul, Etats-Unis, 1988.

[Peled 93] D. PELED, « All from One, One for All: on Model Checking Using Representatives », *Proceedings of the 5th International Conference on Computer Aided Verification (CAV'93)*, p. 409–423, Springer-Verlag, 1993.

[Pencolé et al. 01a] Y. PENCOLÉ, M.-O. CORDIER et L. ROZÉ, « A decentralized model-based diagnostic tool for complex systems », *thirteen IEEE international conference on tools with artificial intelligence (ICTAI'01)*, p. 95–102, Dallas, TX, Etats-Unis, 2001.

[Pencolé et al. 01b] Y. PENCOLÉ, M.-O. CORDIER et L. ROZÉ, « Incremental decentralized diagnosis approach for the supervision of a telecommunication network », *Proceedings of the International Workshop on Principles of Diagnosis (DX'01)*, p. 151–158, Sansicario, Italie, 2001.

[Pencolé et al. 02] Y. PENCOLÉ, M.-O. CORDIER et L. ROZÉ, « Une stratégie efficace pour une approche décentralisée du diagnostic de systèmes complexes », *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA'02)*, p. 259–267, Angers, France, 2002.

[Pencolé 00] Y. PENCOLÉ, « Decentralized diagnoser approach: application to telecommunication networks », *Proceedings of the International Workshop on Principles of Diagnosis (DX'00)*, p. 185–192, Morelia, Mexique, 2000.

[Pujolle 95] G. PUJOLLE, *Les Réseaux*, Eyrolles, 1995.

[Reiter 92] R. REITER, « A theory of diagnosis from first principles », *Artificial Intelligence*, vol. 32, n° 1, 1987 (repris dans *Readings in Nonmonotonic Reasoning*, M. L. Ginsberg (dir.), Morgan Kaufmann, 1987 ; aussi dans *Readings in Model-Based Diagnosis*, W. Hamscher, L. Console, J. de Kleer (dir.), Morgan Kaufmann,p. 29–48, 1992), p. 57–96.

[Riese 93a] M. RIESE, « Diagnosis of Communicating Systems: Dealing with Incompleteness and Uncertainty », *Proceedings of the 13th International Joint Conference On Artificial Intelligence (IJCAI'93)*, p. 1480–1485, Chambéry, France, 1993.

[Riese 93b] M. RIESE, « Diagnosis of Extended Finite Automata as a Dynamic Constraint Satisfaction Problem », *Proceedings of the International Workshop on Principles of Diagnosis(DX'93)*, p. 60–73, Aberystwyth, Grande-Bretagne, 1993.

[Riese 93c] M. RIESE, *Model-Based Diagnosis of communication protocols*, thèse, École polytechnique fédérale de Lausanne, 1993.

[Rozé et Cordier 98] L. ROZÉ et M-O. CORDIER, « Diagnosing Discrete Event Sytems : An experiment in Telecommunication Networks », *WODES98, Fourth Workshop on Discrete Event Systems*, p. 130–137, Cagliari, Italie, août 1998.

[Rozé et Cordier 02] L. ROZÉ et M-O. CORDIER, « Diagnosis of Discrete-Event Systems: Extending the Diagnoser Approach to Deal with Telecommunication Networks », *Discrete Event Dynamic Sytems: Theory and Applications*, vol. 12, 2002, p. 43–81.

[Rozé et Laborie 98] L. ROZÉ et P. LABORIE, « Supervision of Telecommunication Networks : Extending the Classical Diagnoser Approach », *Proceedings of the International Workshop on Principles of Diagnosis (DX'98)*, Cape Cod, Massachussets, Etats-Unis, mai 1998.

[Rozé 97a] L. ROZÉ, *Supervision de réseaux de télécommunication : une approche à base de modèles*, Irisa, Campus de Beaulieu, F-35042 Rennes Cedex, thèse, Ifsic/Université de Rennes 1, 1997.

[Rozé 97b] L. ROZÉ, « Supervision of telecommunication network : a diagnoser approach », *Proceedings of the International Workshop on Principles of Diagnosis (DX'97)*, p. 103–111, Mont St Michel, France, 1997.

[Sampath et al. 95] M. SAMPATH, R. SENGUPTA, S. LAFORTUNE, K. SINNAMOHIDEEN et D. TENEKETZIS, « Diagnosability of Discrete Event System », *IEEE Transactions on Automatic Control*, vol. 40, n° 9, 1995, p. 1555–1575.

[Sampath et al. 98] M. SAMPATH, R. SENGUPTA, S. LAFORTUNE, K. SINNAMOHIDEEN et D. TENEKETZIS, « Active Diagnosis of Discrete-Event Systems », *IEEE Transactions on Automatic Control*, vol. 43, n° 7, 1998, p. 908–929.

[Sengupta 98] R. SENGUPTA, « Diagnosis and communication in distributed systems », *Proceedings of the Workshop on Discrete Event Systems (WODES'98)*, p. 144–151, Cagliari, Italie, août 1998.

[Simony et Znaty 97] N. SIMONY et S. ZNATY, *Gestion de réseau et de service : similitude des concepts, spécificités des solutions*, InterEditions,Masson, 1997.

[sldd 01] B. DUBUISSON (SOUS LA DIRECTION DE), *Diagnostic, intelligence artificielle et reconnaissance des formes*, Hermes, 2001, *Traité IC2 : Information - Commande - Communication*.

[Sloman 89] C. SLOMAN, « A tutorial on OSI management », *Computer Networks and ISDN Systems*, vol. 17, 1989.

[Sloman 94] M. SLOMAN, *Network and Distributed Systems Management*, Addison Wesley, 1994.

[Smyth et al. 91] P. SMYTH, J. STATMAN, G. OLIVER et R. GOODMAN, « Combining knowledge-based techniques and simulation with applications to communications network management », *Integrated Network Management*, I. Krishnan et W. Zimmer ed., p. 505–515, Elsevier Science, 1991.

[Struss et Dressler 92] P. STRUSS et O. DRESSLER, « Physical negation: Integrating fault models into the general diagnostic engine », *Proceedings of the 11th International Joint Conference on Artificial Intelligence IJCAI'89*, p. 1318–1323, Detroit, MI, Etats-Unis, 1989 (repris dans *Readings in Model-Based Diagnosis*, W. Hamscher, L. Console, J. de Kleer (dir.), Morgan Kaufmann,p. 153–158, 1992).

[Thiébaux et al. 94] S. THIÉBAUX, M.-O. CORDIER, O. JEHL et J.-P. KRIVINE, « Supply restoration in power distribution systens – a case study in integrating model-based diagnosis and repair planning », *Proc UAI*, p. 525–532, 1994.

[Turner 93] KENNETH J. TURNER (éd.), *Using Formal Description Techniques — An Introduction to Estelle, LOTOS and SDL*, John Wiley & Sons, 1993.

[Ungauer 93] C. UNGAUER, *Problématique d'utilisation de techniques de supervision à base de connaissances profondes : l'exemple de la supervision du réseau* TRANSPAC, rapport technique, France Telecom R & D, 1993.

[UT 92a] UIT-T, *Synchronous Digital Hierarchy (SDH) Management Information Model for the Network Element View*, 1992.

[UT 92b] UIT-T, *Systems Management — Alarm Reporting Function*, 1992.

[UT 92c] UIT-T, *Systems Management - Event Reporting Management Function*, 1992.

[UT 94] UIT-T, *Synchronous Digital Hierarchy (SDH) Management*, 1994.

[UT 95a] UIT-T, *Generic Network Information Model*, 1995.

[UT 95b] UIT-T, *Principle of a Telecommunication Management Network (TMN)*, 1995.

[UT 96] UIT-T, *Network Node Interface for the Synchronous Digital Hierarchy (SDH)*, 1996.

[Wang 89] Z. WANG, « Model of network faults », *Integrated Network Management*, B. Meandzjia et J. Westcott ed., Elsevier Science, 1989.

[Williams et Nayak 96] B. WILLIAMS et P. NAYAK, « Immobile robots – AI in the new millenium », *AI magazine*, vol. 17(3), 1996.

[Wilsky 76] A.S. WILSKY, « A survey of design methods for failure detection in dynamic systems », *Automatica*, vol. 12, 1976, p. 601–611.

LISTE DES FIGURES

LISTE DES TABLEAUX

Index

Résumé

Le cadre de cette thèse est la surveillance et le diagnostic de systèmes dynamiques complexes tels que les réseaux de télécommunications. Ces systèmes sont composés d'un ensemble d'équipements interconnectés. Des mécanismes liés à des capteurs permettent à un superviseur de recevoir les alarmes émises par tous les composants du réseau et de les interpréter.

L'objectif de ces travaux est de fournir une aide à l'interprétation de ces alarmes afin d'avoir à tout moment une vision de l'état du réseau et de ses dysfonctionnements possibles. L'approche développée est issue des techniques de diagnostic à base de modèles. Elle consiste, à partir d'un modèle du fonctionnement et de dysfonctionnement des composants du réseau, à utiliser efficacement ce modèle pour analyser en-ligne le flux d'alarmes. Dans le cadre de la supervision de tels systèmes, le diagnostic consiste non seulement à établir à partir des observations les états possibles du système à un instant donné mais aussi la propagation des pannes. Nous proposons de représenter ces diagnostics sous forme de systèmes de transitions compacts (transducteurs réduits) basés sur des événements de pannes.

Étant données la complexité et la nature répartie de ces systèmes, nous avons concentré notre étude sur l'élaboration d'une « approche décentralisée » de diagnostic, fondée sur le principe de « diviser pour régner ». Dans un premier temps, nous établissons un ensemble de diagnostics locaux fondés sur des modèles de comportements locaux. Afin d'assurer l'efficacité de ces calculs, nous nous appuyons sur l'approche proposée par M. Sampath et al. qui consiste à construire hors-ligne une structure de données appelée « diagnostiqueur » qui rend le suivi en-ligne et la production d'un diagnostic local possible.

Dans un deuxième temps, l'obtention du diagnostic du système est établi par fusion des diagnostics locaux. Cette fusion est nécessaire car elle permet de valider ou d'invalider les hypothèses locales de diagnostic. Une stratégie de fusion a été mise en place afin d'assurer l'efficacité de cette fusion.

Cette thèse a été effectuée dans le cadre d'un projet RNRT : le projet Magda. Elle a abouti au développement d'une plate-forme pour le diagnostic décentralisé de systèmes dynamiques. Cette plate-forme nous a permis de valider notre approche sur deux types de réseaux : le réseau Transpac et un réseau SDH.

Mots-clé

Diagnostic à base de modèles, systèmes à événements discrets, intelligence artificielle distribuée, réseaux de télécommunications