



Numéro d'ordre : 40773

Thèse de doctorat de l'Université LILLE 1

École Polytechnique Universitaire de LILLE

École Doctorale Sciences pour l'Ingénieur

LABORATOIRE D'AUTOMATIQUE GÉNIE INFORMATIQUE ET SIGNAL

Spécialité : Automatique et Génie Informatique

présentée par

Wissam KHALIL

intitulée

Contribution à la modélisation graphique de système de systèmes

Soutenue le 2 février 2012

devant le jury composé de

Rapporteurs :

Jean-Pierre CAMPAGNE

Prof. à l'INSA Lyon

Dominique SAUTER

Prof. à l'Université Henri Poincaré, Nancy

Examineurs :

Ali CHARARA

Prof. à l'Université de Technologie de Compiègne

François CHARPILLET

Dir. de recherche à l'INRIA Nancy

Geneviève DAUPHIN-TANGUY

Prof. à l'École Centrale de LILLE

Hafid HAFFAF

Prof. à l'Université d'Oran

Directeurs de thèse :

Rochdi MERZOUKI

Prof. à Polytech'Lille

Belkacem OULD-BOUAMAMA

Prof. à Polytech'Lille

Université LILLE 1 Sciences et Technologies
École Polytechnique Universitaire de LILLE
École Doctorale Sciences pour l'Ingénieur
Laboratoire d'Automatique Génie Informatique et Signal
Équipe Méthodes et Outils pour la Conception Intégrée de Systèmes
Cité Scientifique
Boulevard Paul LANGEVIN
59655 VILLENEUVE D'ASCQ
FRANCE.

Au LIBAN, ma patrie ...

*À mes parents
À Zaynab et Laya*

Remerciement

Trouver les bons mots pour dire « Merci » est une chose difficile, d'autant plus dans une langue qui n'est pas sa langue maternelle. Je tiens tout d'abord à manifester ma plus profonde et sincère reconnaissance envers mes directeurs de thèse Monsieur Belkacem OULD-BOUAMAMA et Monsieur Rochdi MERZOUKI pour m'avoir offert la possibilité de réaliser ce travail. Monsieur OULD-BOUAMAMA a toujours fait preuve d'une totale confiance, d'un soutien et d'une gentillesse permanente à mon égard. Malgré son emploi du temps chargé, il a toujours su trouver du temps à m'accorder aussi bien lors de l'encadrement scientifique que dans ma vie quotidienne par ses conseils et suggestions, ce qui a fait nos rapports simples et chaleureux. Les nombreux échanges scientifiques avec Monsieur MERZOUKI m'ont permis de découvrir et apprécier un champ d'investigation nouveau pour moi. Sa rigueur scientifique et ses conseils avisés m'ont aidé et motivé dans la réalisation de cette thèse. Tout deux ont su m'orienter et m'apporter l'optimisme et la confiance en moi nécessaire à la réalisation d'un tel projet et plus particulièrement dans les moments difficiles, où le doute m'effleurait. Pour tout cela, merci.

Mes remerciements vont également à Monsieur Jean-Pierre CAMPAGNE et Monsieur Dominique SAUTER qui m'ont fait l'honneur de rapporter mon travail. Qu'ils soient assurés de ma profonde reconnaissance. Je remercie également Madame Geneviève DAUPHIN-TANGUY, Monsieur Ali CHARARA, Monsieur François CHARPILLET, et Monsieur Hafid HAFFAF d'accepter d'être examinateurs de cette thèse.

La présente thèse a pu commencer grâce à une bourse de mobilité à la recherche, qui m'a été allouée par le programme ERASMUS, auquel j'adresse ma profonde reconnaissance. Mes sincères remerciements s'adressent, en particulier, à Monsieur Hussein MROUEH qui a montré un réel intérêt à mon dossier. Je remercie aussi Madame Véronique LEVEL pour son dévouement à faciliter les démarches administratives des boursiers et ses efforts à rendre leur séjour agréable.

Pendant les années de thèse, j'ai beaucoup apprécié l'ambiance amicale qui règne au Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS), je tiens à remercier tous ses membres et plus particulièrement ceux de l'équipe Méthodes et Outils pour la Conception Intégrée de Systèmes (MOCIS). Je remercie également les secrétaires, notamment Madame Marcelle DESFROMONT pour sa gentillesse. Je suis aussi reconnaissant envers Monsieur Michel POLLART d'avoir répondu à tous les problèmes d'ordre informatique.

Mes remerciements vont également à Madame Anne-Lise GEHIN pour le temps qu'elle m'a accordé et pour les remarques précieuses qu'elle m'a suggérées.

Le plaisir que j'ai eu de faire cette thèse au sein d'un laboratoire remarquablement motivé a été accru par l'existence du projet européen : Intelligent Transportation for Dynamic Environment (InTraDE). Il m'a permis de travailler dans des conditions confortables et de participer à de nombreux congrès. Grâce à cela j'ai pu mieux appréhender le monde maritime. Merci donc, à tous les membres du projet InTraDE pour leur support et l'aiguillon qu'ils ont représenté.

Merci aussi aux doctorants ainsi qu'aux post-doctorant du LAGIS avec qui j'ai pu partager les joies et des difficultés de la vie de thésard, en particulier Nizar CHATTI, Youcef TOUATI et Shahin GELAREH.

Je tiens également à transmettre l'expression de mon plus profond respect à Madame Colette GUILLOPÉ qui a conduit mes premiers pas en FRANCE.

Je n'oublie pas de remercier ma deuxième famille : ma belle-mère, mon beau-père, mes belles-sœurs et mon beau frère et leurs familles respectives.

Cette thèse a également bénéficié du soutien total de ma famille. Je pense aujourd'hui avec beaucoup de tendresse à mes parents, à la confiance qu'ils m'accordent. Je veux les remercier de tout mon cœur, ainsi que mes sœurs et mes frères. Sans votre soutien et votre amour, tout ceci n'aurait jamais pu aboutir et je vous serai éternellement reconnaissant d'avoir su me donner les moyens d'arriver jusqu'ici. Aucun mot ne saurait décrire ma pensée !

Merci aussi à tous les « anonymes » qui auraient certainement mérité d'être cités.



Acronyme

- AGV** Véhicule guidé et automatisé connu sous l'expression anglaise Automated Guided Vehicle
- CSP** Problème de satisfaction de contraintes connu sous l'expression anglaise Constraint Satisfaction Problem
- FDI** Détection et isolation des défauts connu sous l'expression anglaise Fault Detection and Isolation
- IAV** Véhicule intelligent autonome connu sous l'expression anglaise Intelligent Autonomous Vehicle
- ICT** Technologie de l'information et de la communication connu sous l'expression anglaise Information and Communication Technologies
- InTraDE** Intelligent Transportation for Dynamic Environment
- ITS** Système de transport intelligent connu sous l'expression anglaise Intelligent Transportation System
- MGD** Modèle de Graphe Dynamique
- RRAs** Relations de Redondances Analytiques
- SdS** Systèmes de Systèmes
- V2I** Véhicule à l'infrastructure connu sous l'expression anglaise Vehicle to Infrastructure
- V2V** Véhicule à véhicule connu sous l'expression anglaise Vehicle to Vehicle



TABLE DES MATIÈRES

1	Introduction	17
1.1	Contexte général	17
1.2	Problématique et positionnement Scientifique	19
1.2.1	Problématique	19
1.2.2	Contribution	20
1.2.3	Contexte du travail de thèse au sein de l'équipe	21
1.3	Organisation de la thèse	21
2	Modélisation des systèmes de systèmes : état de l'art	23
2.1	Introduction	24
2.2	Système de systèmes	25
2.2.1	Définition	25
2.2.2	Historique et applications des systèmes de systèmes	27
2.3	Formalismes de modélisation	29
2.4	Formalismes graphiques	30
2.4.1	Réseau sémantique	30
2.4.2	Réseau bayésien	32
2.4.3	Réseau de Petri	33
2.4.4	Digraphe	34
2.4.5	Graphe fonctionnel	35
2.4.6	Graphe biparti	35
2.4.7	Bond graph	36
2.4.8	Graphe	37
2.4.9	Hypergraphe	45
2.5	Conclusion	46

3	Modélisation des systèmes de systèmes par l'approche de l'hypergraphe	49
3.1	Introduction	50
3.2	Représentation multi-niveaux et multi-structures d'un système de systèmes	51
3.3	Modélisation d'un système de systèmes par l'approche de l'hypergraphe	53
3.4	Propriétés du système de systèmes à partir du modèle de l'hypergraphe	59
3.5	Supervision d'un système de systèmes à base d'un modèle hypergraphe	64
3.5.1	Architecture du système de supervision	64
3.5.2	Surveillance	65
3.5.3	Gestion des modes de fonctionnement sur l'hypergraphe	66
3.5.4	Parcours descendant de l'hypergraphe pour la reconfiguration	71
3.6	Application à un système de transport intelligent	74
3.6.1	Modélisation du système de transport intelligent	76
3.6.2	Surveillance	79
3.6.3	Identification des modes de fonctionnement	80
3.6.4	Reconfiguration	84
3.7	Conclusion	85
4	Modélisation des systèmes de systèmes par l'approche du graphe dynamique	87
4.1	Introduction	88
4.1.1	Contexte scientifique	88
4.1.2	Contexte applicatif	89
4.2	Passage du modèle hypergraphe à un modèle de graphe dynamique	90
4.3	Modèle de graphe dynamique d'un système de systèmes	92
4.4	Optimisation en ligne sur un modèle de graphe dynamique	94
4.4.1	Extraction du graphe statique \mathfrak{N}_t	95
4.4.2	Décomposition de \mathfrak{N}_t et \mathcal{G}_t en sous-graphes	96
4.4.3	Valuation des arcs du graphe statique \mathfrak{N}_t	98
4.5	Application à un système de transport intelligent	98
4.6	Résultats de Simulation	105
4.6.1	Description du terrain	105
4.6.2	Scénario de routage	105
4.7	Conclusion	112

5 Conclusion et perspectives	113
5.1 Conclusion	113
5.2 Perspectives	114
A Simulation 3D - temps réel	119
A.1 Pourquoi la simulation 3D - temps réel?	119
A.2 Rôle du simulateur	120
A.3 Conception d'un scénario de simulation en temps réel	121
A.3.1 Mode terrain	121
A.3.2 Mode véhicule	122
A.3.3 Mode scénario	122
A.3.4 Mode simulation	122
A.3.5 Mode analyse	124
A.4 Implémentation de modules externes sur le simulateur	124
B Liste des publications scientifiques	127
B.1 Article de journal	127
B.2 Articles de conférences internationales avec comité de lecture	127

TABLE DES FIGURES

1.1	WP3 : supervision du système ITS	18
2.1	Formalismes de modélisation suivant [LJUNG, 1999]	29
2.2	Formalismes graphiques	31
2.3	Réseau sémantique	31
2.4	Réseau bayésien	32
2.5	Réseau de Petri	33
2.6	Digraphe	35
2.7	Graphe fonctionnel	35
2.8	Graphe biparti	36
2.9	Bond graph	37
2.10	(a) Ponts de Königsberg, (b) Problème du cavalier, (c) Problème de coloriage de cartes, (d) Problème hamiltonien.	38
2.11	Graphe non-orienté composé de cinq sommets et quatre arêtes.	39
2.12	Graphe orienté composé de cinq sommets et six arcs.	40
2.13	Représentation du graphe non-orienté de la FIGURE 2.11 par un graphe orienté.	40
2.14	Graphe orienté valué.	40
2.15	Un sous-graphe du graphe orienté de la FIGURE 2.12.	41
2.16	Le graphe et ses dérivés	42
2.17	Exemple d'un graphe évolutif dynamique (SG) construit à partir des trois graphes statiques (G_1 , G_2 et G_3) représentant des graphes évolutifs.	44
2.18	Hypergraphe composé de huit sommets et six hyperarêtes.	45
2.19	(a) graphe et (b) hypergraphe.	46
3.1	Exemple d'un modèle multi-niveaux et multi-structures appliqué au transport	51

3.2	Modèle multi-niveaux et multi-structures	53
3.3	Hyperensemble \mathcal{HE}	54
3.4	Hyperensemble imbriqué	55
3.5	Représentation ensembliste d'un hypergraphe composé de sept sommets et cinq hyperarêtes interconnectées.	56
3.6	Hypergraphe imbriqué	57
3.7	Hypergraphe dual de celui de la FIGURE 3.5.	57
3.8	Hypergraphe \mathcal{HG} , à gauche : représentation ensembliste, à droite : graphe valué.	58
3.9	Hypergraphe valué de la FIGURE 3.5.	59
3.10	Indépendance opérationnelle	60
3.11	Indépendance managériale	60
3.12	Dispersion géographique	61
3.13	Comportement émergent et coopératif	62
3.14	Développement évolutionnaire	62
3.15	Représentation du système $CS_{1,3}$ par un hyperensemble (à gauche) et par un hypergraphe imbriqué et valué (à droite) . .	63
3.16	représentation hiérarchique de l'hypergraphe imbriqué et valué de la FIGURE 3.15	63
3.17	Modes de fonctionnement du système $CS_{1,2}$	68
3.18	Représentation hiérarchique d'un hypergraphe imbriqué et valué	69
3.19	Influence de $CS_{4,1}$ sur $CS_{1,2}$ et $CS_{1,1}$	70
3.20	Propagation de l'effet de la défaillance sur le SdS	71
3.21	Reconfiguration du système $CS_{1,3}$ sans le composant système $CS_{4,1}$	73
3.22	Vue aérienne du terminal maritime Radicatel	75
3.23	Train de trois véhicules intelligents autonomes	75
3.24	Composant système IAV	76
3.25	Hypergraphe imbriqué et valué d'un ITS : représentation hiérarchique	77
3.26	Hypergraphe imbriqué et valué d'un ITS : représentation ensembliste	78
3.27	Modes de fonctionnement des actionneurs	81
3.28	Modes de fonctionnement du train de IAVs	82
3.29	Analyse du mode de fonctionnement de l'actionneur Ac_1^2	83
3.30	ITS en mode « dégradé »	83
3.31	Suppression de IAV_2 du train	84
3.32	Le train après la reconfiguration	85
3.33	Mode de fonctionnement de trois IAVs	86
3.34	Vitesse réelle de trois IAVs	86

4.1	Modèle multi-niveaux d'un SdS	90
4.2	Découplage du niveau j par rapport aux niveaux $j - 1$ et $j + 1$	91
4.3	Modèle de graphe dynamique du niveau j	92
4.4	Exemple de MGD pour un SdS $\mathcal{G}_t(\mathcal{N}, \mathcal{A}, \mathfrak{F}_{\mathcal{G}})$	93
4.5	Extraction du graphe statique \mathfrak{N}_t à partir du graphe dynamique \mathcal{G}_t de la FIGURE 4.4	96
4.6	Décomposition de \mathfrak{N}_t en sous-graphes	97
4.7	Décomposition de \mathcal{G}_t en sous-graphes	97
4.8	Graphe dynamique \mathcal{G}_t du système ITS	98
4.9	Le graphe statique \mathfrak{N}_t	100
4.10	Différents scénarios de routage des IAVs sur une route $R_{i,j}$	103
4.11	La distance $\tilde{d}_{i,V_{k+1}}$	104
4.12	Les deux cas en fonction de $\tilde{d}_{i,V_{k+1}}$	104
4.13	Terminal maritime « Radicatel » sous SCANeR studio	105
4.14	Opération de routage dans le domaine de l'exportation	106
4.15	Le plus court parcours	107
4.16	Le véhicule V_6 ne peut pas atteindre sa destination	108
4.17	Le chemin ayant le nombre minimal de véhicules	109
4.18	Chemin optimal d'après l'algorithme proposé	109
4.19	Atteindre la destination S_2 après 84s	110
4.20	Atteindre la destination S_2 en un temps optimal	111
4.21	Chemin optimal selon les trois méthodes étudiées	112
A.1	SCANeR studio	120
A.2	Mode terrain	121
A.3	Mode véhicule	122
A.4	Mode scénario	123
A.5	Mode simulation	123
A.6	Mode analyse	124
A.7	SCANeR studio et les modules externes	125

CHAPITRE 1

Introduction

Sommaire

1.1	Contexte général	17
1.2	Problématique et positionnement Scientifique . .	19
1.2.1	Problématique	19
1.2.2	Contribution	20
1.2.3	Contexte du travail de thèse au sein de l'équipe . .	21
1.3	Organisation de la thèse	21

1.1 Contexte général

Le commerce maritime mondial n'a cessé de se développer cette dernière décennie, dû principalement à la mondialisation et au développement des pays émergents. La croissance du commerce maritime influe directement sur le développement des ports et terminaux maritimes. La zone NWE (North West Europe) dispose d'un large littoral allant de l'Irlande à l'Allemagne. Malheureusement, peu de ports de petites et moyennes tailles de cette région arrivent à suivre le rythme de cette croissance mondiale. Ces zones et terminaux portuaires affrontent au quotidien des difficultés liées à la gestion du trafic routier et maritime ainsi qu'à l'optimisation de l'espace opérationnel.

Ainsi, le projet européen [InTraDE] a vu le jour en 2009, afin de répondre aux problématiques liées à la gestion de trafic et à l'optimisation de l'espace à l'intérieur des zones confinées des ports à conteneurs. InTraDE (Intelligent

Transportation for Dynamic Environment) est un projet innovant transnational, visant à concevoir un système de transport intelligent (ITS) s'adaptant à l'environnement opérationnel. Ce dernier doit respecter les contraintes socio-économiques, écologiques pour le transport du fret à l'intérieur des terminaux portuaires des régions de la zone Nord Ouest de l'Europe.

Le système ITS proposé dans le cadre de ce projet est composé de deux livrables innovants :

1. Deux prototypes de véhicules autonomes intelligents, porte-conteneurs, omnidirectionnels de conception, pouvant être pilotés manuellement ou automatiquement, de façon individuelle ou en train de véhicules ;
2. Un simulateur 3D opérant en temps réel, reprenant la dynamique des véhicules dans un environnement graphique mixé avec une reconstruction de terrain et d'environnement. Cet outil est utilisé à la fois pour la conception d'un système ITS et pour la supervision en ligne de tous ses composants systèmes.

Ainsi, le travail de thèse présenté dans ce manuscrit rentre dans le cadre de l'action « Supervision du système de transport intelligent », et plus particulièrement dans les tâches modélisation et identification (WP3A5), et gestion de trafic et optimisation (WP3A7) comme le montre la FIGURE 1.1.

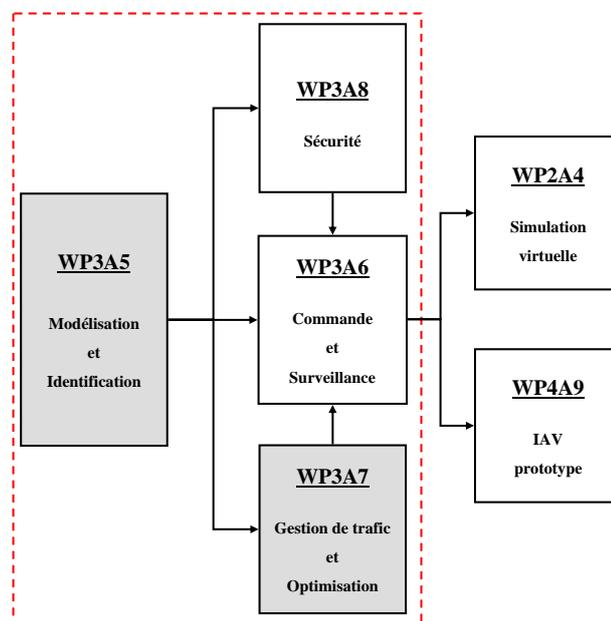


FIGURE 1.1 – WP3 : supervision du système ITS

1.2 Problématique et positionnement Scientifique

1.2.1 Problématique

De nos jours, les systèmes d'ingénierie sont de plus en plus complexes, non seulement par leur nature multi-physique mais aussi par leur opérabilité dans un environnement variable et d'autant plus complexe. Ainsi, ces systèmes sont confrontés à la nécessité d'envisager l'intégration des enjeux humains, organisationnels et technologiques dans toutes les phases du cycle de vie, indispensables à leur mise en service quotidienne. Ces systèmes d'ingénierie, pouvant coopérer de façon indépendante afin de réaliser des missions globales suivant le changement permanent de l'état de l'environnement qui les entoure, peuvent être décrits sous forme d'un concept connu sous le nom de Système de Systèmes (SdS).

Peu d'équipes de recherche au niveau national travaillent dans ce domaine. Le laboratoire le plus reconnu dans cette thématique est le laboratoire Heudiasyc (Heuristique et Diagnostic des Systèmes Complexes) de l'Université de Technologie de Compiègne (UTC) dirigé par le professeur Ali CHARARA. Ce laboratoire est associé à un réseau du laboratoire d'excellence Labex dans la Maîtrise des Systèmes de Systèmes Technologiques (MS2T). Visant un large champ d'applications, la MS2T répond à de forts enjeux socio-économiques dans les domaines du transport et de la mobilité (voiture électrique intelligente, systèmes de transports multimodaux), de la sécurité (mini-drones), de l'ingénierie pour la santé (rééducation fonctionnelle, micro-nano technologies pour les systèmes biologiques) et de l'environnement (gestion en temps réel de l'évacuation des eaux pluviales) [CHARARA, 2011].

Au niveau international, plusieurs équipes de recherche sont impliqués dans ce domaine. Nous citons par exemple, le département « Electrical and Computer Engineering » de l'université de Texas [JAMSHIDI, 2008], le département « Systems Engineering and Operations Research » de l'université George Mason [SAGE, 2001], le département « Electrical and Computer Engineering » de l'université de New Mexico [SRIDHAR, 2007],

Un SdS se définit comme un nouveau concept organisationnel de plusieurs entités indépendantes, inter-communicantes avec une hétérogénéité fonctionnelle, évoluant de façon dynamique suivant le mode de fonctionnement de chaque entité et les changements de l'environnement encapsulant. Un sys-

tème ITS peut être conceptualisé sous la forme d'un SdS composés d'entités (composants systèmes) dispersées géographiquement, indépendantes d'un point de vue opérationnel et managérial mais devant collaborer pour réaliser une tâche globale et évolutive.

La structure d'un tel concept pose le problème de sa modélisation en vue de sa représentation, son analyse mais aussi pour assurer son pilotage. En effet, un SdS fait apparaître plusieurs niveaux hiérarchiques de modélisation selon le niveau microscopique ou macroscopique utilisé pour la modélisation des composants systèmes qu'il regroupe.

Dans un SdS, les systèmes échangent des flux d'informations (entre les différents composants systèmes) mais aussi des flux d'énergies et de signaux physiques (entre les sous-systèmes élémentaires d'un même composant système).

1.2.2 Contribution

Peu de travaux dans la littérature concernent des outils structurés de modélisation des SdS dont la notion, il est vrai, n'a été définie que vers les années 90. La plupart des travaux (comme détaillé dans l'état de l'art au chapitre 2) se contente de descriptions générales et concerne principalement des applications militaires et de gestion. Notre contribution scientifique, à travers ce travail de recherche, concerne :

1. **Développement de modèles graphiques d'un SdS** : les modèles graphiques d'un SdS sont représentés sur plusieurs niveaux et respectant ses propriétés fondamentales à savoir : l'indépendance opérationnelle et managériale, la coopération, la dispersion et le développement évolutionnaire. Deux modèles graphiques ont été proposés pour deux objectifs distincts : la supervision et l'optimisation.
2. **Exploitation du modèle SdS pour la supervision** : le premier est basé sur l'approche de l'hypergraphe, où la notion de l'hyperarête est introduite afin de généraliser la liaison graphique (arête) entre deux nœuds par une liaison multi-nœuds et multi-niveaux. Il a été démontré dans ce travail de recherche, qu'à travers le modèle de l'hypergraphe, les propriétés d'un SdS sont préservées. Une décomposition hiérarchique d'un SdS sous forme d'hypergraphes va nous permettre d'élaborer une stratégie de supervision. En fait, le parcours ascendant de l'hypergraphe permet de suivre les conséquences d'un changement de mode de fonctionnement d'un composant système, à un niveau hiérarchique donné,

sur l'ensemble des composants systèmes de niveau supérieur et ainsi de détecter le mode de fonctionnement global du SdS. A travers le parcours descendant, une optimisation verticale (multi-niveaux) est réalisée afin de définir des reconfigurations sur le SdS, permettant de maintenir la réalisation de sa mission globale avec des performances identiques ou inférieures.

3. **Exploitation du modèle SdS pour l'optimisation** : le deuxième modèle graphique de SdS est dérivé du modèle de l'hypergraphe mais se focalise sur un niveau particulier de modélisation. Ce modèle dynamique et évolutif dépendant du temps, permet d'élaborer des algorithmes d'optimisation en ligne pour les composants systèmes associés à un niveau de modélisation. Pour ce type d'optimisation dite transversale, un modèle multi-niveaux par hypergraphe n'est pas adéquat. En effet, les interactions entre ce niveau de modélisation et les autres niveaux ont besoin d'être représentées par des poids de fonctions agissant directement sur les nœuds et les arcs. Ce modèle a été exploité pour le calcul du temps optimal et adaptatif, qu'un véhicule intelligent autonome (IAV) prend pour relier un nœud de départ et un nœud de destination dans un réseau évolutif d'un système ITS.

1.2.3 Contexte du travail de thèse au sein de l'équipe

Ce travail a été réalisé au sein de l'équipe « Méthodes et Outils pour la Conception Intégrée de Systèmes (MOCIS) » du Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS). Cette équipe possède une grande expérience dans la modélisation par bond graph et graphes de systèmes multiphysiques, hybrides, incertains et hamiltoniens, appliquée aux systèmes de transport, à la robotique et aux systèmes énergétiques. Forte de ces compétences, l'équipe MOCIS s'intéresse à la modélisation graphique de SdS, une recherche qui a trouvé une valorisation significative avec le lancement du projet d'InTraDE en 2009. Cette thématique de recherche a fait l'objet, à ce jour, de six communications dans des conférences internationales et un article de journal.

1.3 Organisation de la thèse

Après le chapitre d'introduction générale, trois autres chapitres constituent ce manuscrit de la façon suivante :

Dans le chapitre 2, nous présentons un état de l'art sur le SdS, en définissant

ses propriétés fondamentales et ses domaines d'application. Les formalismes de modélisation les plus utilisés sont décrits dans ce chapitre, en particulier l'hypergraphe, utilisé pour la modélisation de SdS.

Dans le chapitre 3, nous présentons la modélisation d'un SdS par l'approche de l'hypergraphe. La représentation multi-niveaux et la définition de l'hyperensemble sont décrits, permettant d'étendre l'approche d'hypergraphe à la modélisation multi-niveaux de SdS. Ce modèle est ensuite exploité pour l'élaboration d'une stratégie de supervision appliquée à un système ITS.

Dans le chapitre 4, nous présentons le modèle graphique dynamique d'un SdS sur un niveau de modélisation, après une extraction du modèle multi-niveaux de l'hypergraphe. Une procédure d'optimisation en ligne appliquée aux composants systèmes sur le modèle graphique dynamique, permet de définir un temps optimal dans le cas du routage des véhicules IAVs dans un système ITS.

Enfin, dans le dernier chapitre, nous concluons sur notre contribution et nous citons les perspectives futures.

CHAPITRE 2

Modélisation des systèmes de systèmes : état de l'art

Sommaire

2.1	Introduction	24
2.2	Système de systèmes	25
2.2.1	Définition	25
2.2.2	Historique et applications des systèmes de systèmes	27
2.3	Formalismes de modélisation	29
2.4	Formalismes graphiques	30
2.4.1	Réseau sémantique	30
2.4.2	Réseau bayésien	32
2.4.3	Réseau de Petri	33
2.4.4	Digraphe	34
2.4.5	Graphe fonctionnel	35
2.4.6	Graphe biparti	35
2.4.7	Bond graph	36
2.4.8	Graphe	37
2.4.9	Hypergraphe	45
2.5	Conclusion	46

2.1 Introduction

Un modèle est considéré comme une abstraction de la réalité, il décrit au mieux la réalité sans pouvoir la reconstituer parfaitement [MULLER, 2000]. Ainsi, le modèle se caractérise par :

- une représentation fidèle du système à modéliser,
- une simplification de la complexité du système à modéliser,
- une focalisation sur les propriétés importantes du système en négligeant certains détails inutiles,
- une simulation des différents comportements du système à modéliser,
- une traduction du comportement qui coûte moins cher que le teste du comportement réel.

Généralement, le modèle associé à un système permet de :

- prendre en compte toute la complexité du système en la représentant le plus fidèlement possible, grâce à la capitalisation des connaissances sur le système,
- représenter de manière fidèle le comportement du système, avec la possibilité de prédire son évolution autour d'un point de fonctionnement optimal.

Sachant que la précision du modèle dépend du niveau de connaissance du système, une représentation verticale permet d'associer à chaque niveau la modélisation correspondante. Ainsi, lorsque nous souhaitons décrire le comportement d'un système au niveau microscopique, nous parlons de la micro-modélisation, au niveau mésoscopique, nous étudions la méso-modélisation et pour le niveau macroscopique, nous associons la macro-modélisation.

Sur le plan horizontal, il est nécessaire que les systèmes puissent également interagir entre eux. Cette interaction peut se présenter sous forme de co-existences ou de coopérations. Dans le cas où ils coexistent, chaque système considère les autres systèmes comme étant des composantes de l'environnement. Chaque système agit alors d'une façon autonome et flexible pour réaliser les objectifs pour lesquels il a été conçu. Néanmoins, dans le cas où ils coopèrent, les systèmes effectuent non seulement leurs propres tâches, mais doivent également être apte à interagir entre eux dans le but de réaliser un objectif global. On parle alors de Système de Systèmes (SdS).

Le modèle d'un tel SdS est plus difficile à obtenir, car il doit prendre en considération l'état de chaque composant système et son interaction avec l'ensemble des autres composants de façon dynamique.

À travers ce chapitre, nous allons définir le principe et les propriétés d'un SdS permettant d'introduire notre contribution dans l'élaboration d'un modèle graphique associé, évoluant de façon dynamique. Un positionnement de nos résultats de recherche dans la modélisation graphique de SdS par rapport aux différents formalismes de modélisation est présenté, permettant ainsi d'introduire les deux chapitres suivants sur la modélisation de SdS, par l'approche de l'hypergraphe et par l'approche du graphe dynamique.

2.2 Système de systèmes

2.2.1 Définition

Plusieurs définitions sont citées dans la littérature pour décrire un SdS. Malheureusement, aucune d'entre elles n'est universellement acceptée. Selon [SRIDHAR, 2007], ces définitions ne distinguent pas les SdS des systèmes qui ne le sont pas. En effet, d'un point de vue formel, tout système moderne, peut de nos jours être vu comme un SdS. À titre d'exemples, on peut se demander si :

un véhicule routier de motorisation électrique et composé de plusieurs sous systèmes interconnectés, effectuant une ou plusieurs tâches, est-il un SdS ? le réseau Web composé d'un ensemble non fini de pages Internet hyper-liées est-il un SdS ?

Sans connaître les caractéristiques d'un SdS, ces deux exemples décrivent un ensemble de systèmes co-existants et coopératifs, où pour le véhicule routier le système reste monolithique¹ par rapport à son objectif, et pour le réseau Web, il reste un système évolutif, non complètement formé initialement. En plus, nous remarquons sur ces deux exemples une distinction par rapport aux indépendances managériale et opérationnelle de chaque composant « système » de l'ensemble de SdS.

À ce jour, avec l'évolution de la technologie, les systèmes modernes s'éloignent de plus en plus de leur configuration monolithique. Maier a attribué cinq caractéristiques aux systèmes modernes [MAIER, 1996]. Elles ont permis de définir le concept de SdS [KRYGIEL, 1999], [CARLOCK, 2001], et ont été ap-

1. Un système monolithique est un système qui centralise l'ensemble des informations et tâches nécessaires pour son fonctionnement sur un même support. Exemple, une architecture logicielle monolithique traite à la fois les données et les interfaces homme-machine sur le même système.

pliquées notamment dans les travaux de [SAGE, 2001]. Ces caractéristiques sont :

1. **Indépendance opérationnelle des systèmes** : Un SdS est composé de systèmes qui opèrent indépendamment. Si un SdS est désassemblé en un ensemble de systèmes, ces derniers sont capables de remplir les opérations utiles indépendamment les uns des autres. Ainsi, chaque système possède son propre mode opératoire qui est indépendant des autres systèmes. Par exemple, un ensemble de Véhicules Intelligents Autonomes (IAVs) opérant dans un terminal portuaire, réalise une tâche commune qui consiste à acheminer du fret du point *A* vers un point *B*. Chaque IAV opère indépendamment et séparément des autres IAVs.
2. **Indépendance managériale des systèmes** : Chaque composant système gère individuellement et de façon continue le fonctionnement de ses sous-systèmes, afin de maintenir une exploitation indépendante du SdS. Exemple, un IAV qui opère indépendamment dans un SdS, réalise des missions locales indépendantes aux autres IAVs et qui répondent à la mission globale du SdS.
3. **Dispersion Géographique** : La dispersion géographique des composants systèmes est essentielle. Ces derniers peuvent facilement échanger différents types d'informations entre eux, sauf des énergies physiques.
4. **Comportement émergent et coopératif** : Le SdS exécute des fonctions coopératives et réalise des tâches précises qui ne résident pas dans n'importe quel composant système. Ce comportement est une propriété émergente de l'ensemble du SdS et non pas le comportement d'un composant système. Les principaux objectifs réalisés par un SdS sont satisfaits par ce comportement émergent. L'acheminement d'un certain nombre de conteneurs en un temps prédéfini par des IAVs en configuration de train de véhicules purement automatisé et semi-automatique représente un exemple d'un comportement émergent de l'ensemble de SdS d'IAVs qui ne pourra pas être réalisé par un composant système.
5. **Développement évolutif** : Un SdS n'est jamais entièrement formé et complet. Le développement de ces systèmes reste évolutif dans le temps suivant la structure, la fonction, l'ajout ou la suppression de composants systèmes. Le dimensionnement d'un système d'IAVs évolue au fil du temps dans un environnement d'exploitation dynamique. L'association des infrastructures, du transport ferroviaire, des opérateurs humains ainsi que des IAVs dans l'espace du port peut constituer une première version d'un SdS. Cette dernière peut évoluer dans le temps

suivant les objectifs attendus en rajoutant des composants systèmes de type ICT ou en supprimant d'autres composants systèmes de type IAV.

Étant donné les caractéristiques fondamentales d'un SdS, il devient possible de définir un SdS dans le contexte de notre travail, en le différenciant d'un système non SdS de la façon suivante :

Définition 2.1 : *Un SdS est un concept décrivant un ensemble de composants systèmes, hétérogènes et indépendants aux échelles managériale et opérationnelle, pouvant coexister et coopérer afin d'accomplir des objectifs, qu'un composant système ne pourra pas réaliser. Ces composants sont dispersés géographiquement sans liaison physique commune. Du SdS ainsi réalisé, des composants systèmes peuvent être ajoutés ou supprimés sans toutefois modifier les caractéristiques du SdS résultant.*

Prenons l'exemple du véhicule routier introduit précédemment. Ce véhicule possède un moteur électrique principal, qui coexiste avec d'autres composants. Si ce moteur est hors fonctionnement, le véhicule ne sera plus opérationnel. Cette dépendance opérationnelle ne permet pas d'atteindre l'objectif fixé à ce véhicule routier, à savoir le transport de manière générale. Ainsi, ce système ne peut pas être considéré comme un SdS. Concernant l'exemple des pages Web, ces dernières possèdent souvent des adresses distinctes. Elles sont donc complètement indépendantes, dispersées géographiquement et évolutives dans le temps. Ces pages coexistent mais ne coopèrent pas automatiquement. Si cette coopération est mise en œuvre, l'ensemble du système Web peut être considéré comme un SdS.

2.2.2 Historique et applications des systèmes de systèmes

Bien que l'introduction du concept de SdS est présente dans les travaux de [BOULDING, 1956], [ACKOFF, 1971], [JACOB, 1974], et [JACKSON, 1984], la notion de « *système de systèmes* » n'a été définie qu'à partir des années 90. Il est clair que le concept de SdS est maintenant omniprésent dans de nombreux systèmes d'ingénierie et de gestion [SRIDHAR, 2007]. On peut citer, entre autres les travaux suivants :

- [EISNER, 1991] a décrit le SdS dans le domaine de l'ingénierie comme un ensemble de systèmes indépendants, qui réalisent une solution multifonctions pour une mission globale cohérente. L'optimisation de chaque système ne garantit pas l'optimisation de l'ensemble de SdS.

- [SHENHAR, 1994] a décrit le SdS comme un réseau de systèmes indépendants, qui fonctionnent ensemble afin de réaliser une tâche commune.
- [HOLLAND, 1995] a comparé le SdS à un système complexe adaptatif, capable de s'auto-adapter avec son environnement. Cette approche est utilisée aussi par [OWENS, 1995] pour décrire les applications de SdS dans le domaine militaire.
- [MANTHORPE, 1996] a proposé de combiner les aspects de commande, de communication, de calculateurs, d'intelligence artificielle, de surveillance et de reconnaissance pour décrire un SdS militaire. Durant la même période, [MAIER, 1996] a introduit les caractéristiques d'un SdS.
- [PEI, 2000] a introduit une extension de SdS, appelée système de systèmes d'intégration, incluant en plus l'optimisation des systèmes.
- En 2001, le département d'état Américain de la défense a publié un rapport [JCIDS, 2001], où il compare un SdS comme une famille de systèmes.
- [BAR-YAM, 2004] a étudié le SdS dans différents domaines applicatifs en comparant leurs caractéristiques.
- [DELAURENTIS, 2005] a décrit le SdS dans le domaine du transport.
- [BOARDMAN, 2006] a comparé plus de 40 définitions de SdS pour les résumer en cinq mots clés : autonomie, appartenance, connectivité, diversité, et émergence.
- En 2008-2010, on a vu l'apparition d'ouvrages consacrés aux SdS tels que : [LUZEAUX, 2008-a], [LUZEAUX, 2008-b], [JAMSHIDI, 2008], [CANTOT, 2009], [JAMSHIDI, 2009], et [NANAYAKKARA, 2010]. Ces travaux consistent à définir le concept des SdS et de l'ingénierie des SdS en présentant quelques illustrations pratiques.

Initialement, les applications de SdS étaient orientées vers des applications militaires dans les systèmes d'ingénierie [DAHMAN, 2008], [ARMY, 2003], [KRYGIEL, 1999]. Actuellement, ce concept s'est étendu à d'autres domaines tels que : l'éducation [LUSKASIK, 1998], le transport [DELAURENTIS, 2005], [DELAURENTIS, 2006], la sécurité [WALKER, 2007], l'observation terrestre [GLOBAL, 2007], la gestion de l'environnement [HIPEL, 2008], l'énergie re-

nouvelable [DUFFY, 2008], le service [TIEN, 2008], la santé [SHIBASAKI, 2008], [WICKRAMASINGHE, 2008]

La sécurité est l'un des domaines où la notion de SdS s'applique de plus en plus. Il est communément admis que la sécurité à grande échelle ne peut pas se représenter avec les outils et les méthodes utilisées pour décrire un simple système [WALKER, 2007].

Toujours, dans les systèmes d'ingénierie, lorsque nous possédons une architecture répondant aux caractéristiques de SdS, il devient important de la modéliser afin d'élaborer des schémas de contrôle/commande, de surveillance, de supervision, et de maintenance. Actuellement, l'absence de modèles descriptifs du comportement d'un SdS, rend difficile l'exploitation de ce concept dans le développement d'algorithme de supervision ou d'optimisation. Ainsi, notre problématique de recherche dans le cadre de cette thèse est formulée autour de la modélisation graphique du concept de SdS, en considérant la dimension du temps et la dynamique évolutive de l'ensemble des composants systèmes.

2.3 Formalismes de modélisation

Plusieurs formalismes sont utilisés, dans la littérature, pour représenter un modèle. Dans [LJUNG, 1999] les formalismes de modélisation sont classés en quatre catégories (FIGURE 2.1) :

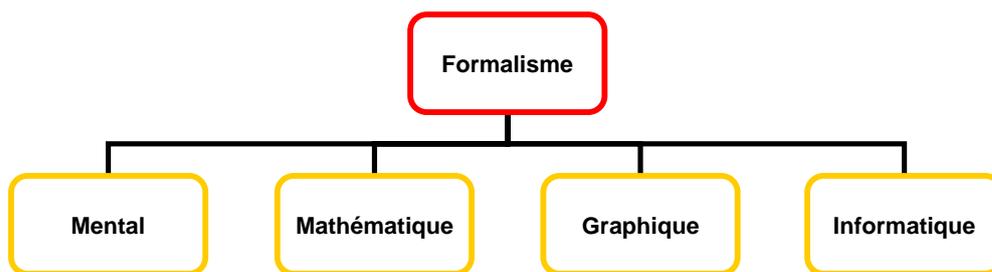


FIGURE 2.1 – Formalismes de modélisation suivant [LJUNG, 1999]

- les formalismes mentaux permettent de simuler mentalement le déroulement d'un phénomène pour anticiper les résultats d'une action. Ces formalismes sont très utilisés dans la vie quotidienne.

- les formalismes mathématiques décrivent les relations entre les variables physiques par des équations mathématiques comme les équations différentielles. Ces formalismes sont utilisés dans presque tous les domaines de la science. Ils nécessitent une connaissance précise et analytique du comportement du système.
- les formalismes graphiques décrivent les systèmes par des graphes. Ils décrivent la structure et l'influence mutuelle entre les variables et/ou les systèmes. Ces formalismes sont bien adaptés pour la modélisation des systèmes complexes à des fins d'analyse ou de synthèse.
- les formalismes informatiques correspondent à des logiciels qui peuvent être constitués de plusieurs sous-programmes interconnectés et de tables de conversions qu'il n'est pas possible de représenter analytiquement. Ces outils informatiques sont utilisés dans le domaine de la prise de décision.

Cette représentation n'est pas une classification figée. On peut avoir un formalisme mixte qui se situe dans plusieurs catégories. Dans la suite de notre développement, nous allons nous intéresser particulièrement au formalisme graphique pour la modélisation de SdS.

2.4 Formalismes graphiques

Plusieurs outils ont été développés pour représenter graphiquement un modèle. Parmi ceux utilisés fréquemment dans le domaine des sciences et technologie de l'information, nous citons : réseau sémantique, réseau bayésien, réseau de Petri, digraphe, graphe biparti, bond graph, graphe, et hypergraphe (FIGURE 2.2).

2.4.1 Réseau sémantique

L'une des particularités de cet outil est de permettre la représentation des connaissances sous une forme graphique (FIGURE 2.3). Le réseau sémantique a été introduit par [QUILLIAN, 1968]. Il est constitué d'un graphe où les sommets représentent des concepts et les arcs orientés lient les concepts entre eux.

La modélisation par le réseau sémantique est utilisée dans plusieurs domaines tels que : l'indexation de documents [GAINES, 1994], la recherche d'information [KHEIRBEK, 1995], l'éducation [GAINES, 1995], [MARTIN, 1996], les

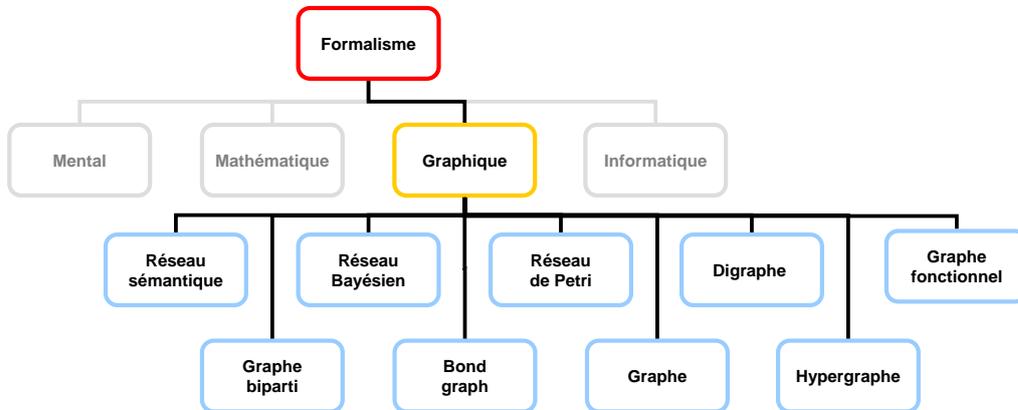


FIGURE 2.2 – Formalismes graphiques

systèmes de traitement automatique du langage [CORNUJOLS, 2002]. Parmi les approches les plus performantes par la recherche d'informations figurent les approches fondées sur les ontologies [DECKER, 1999], [MARTIN, 1999], [VALLET, 2005].

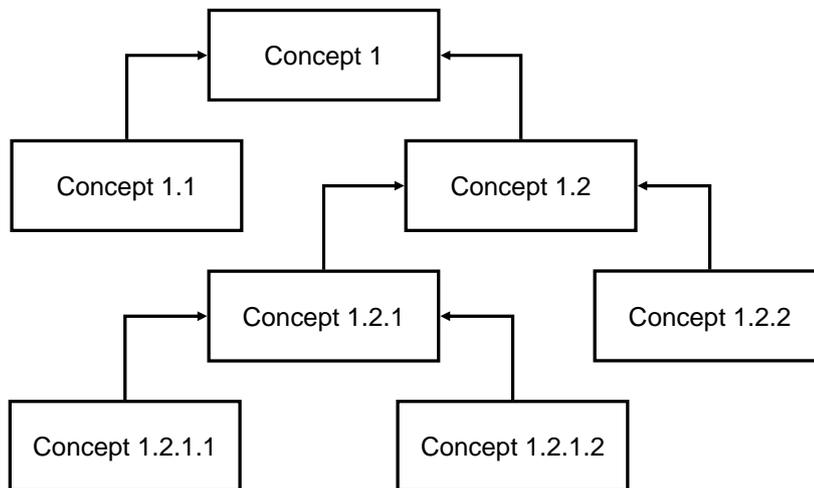


FIGURE 2.3 – Réseau sémantique

2.4.2 Réseau bayésien

Les réseaux bayésiens² sont des formalismes probabilistes graphiques permettant d'acquérir, d'exploiter des connaissances et de calculer des probabilités conditionnelles apportant des solutions à différentes sortes de problématiques [BEN-GAL, 2007].

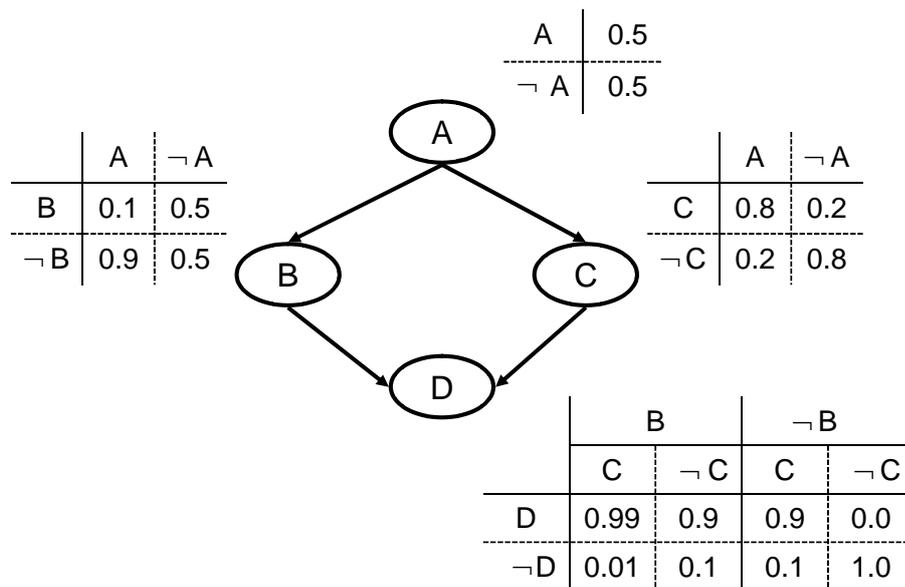


FIGURE 2.4 – Réseau bayésien

Graphiquement, un réseau bayésien est représenté par un graphe orienté acyclique³ (FIGURE 2.4) dont :

- les sommets sont étiquetés par des variables de probabilité,
- les sommets sont reliés entre eux par des arcs orientés,
- chaque sommet contient une table de probabilités conditionnelles.

Le réseau est particulièrement adapté pour la prise en compte de l'incertitude, où la structure peut aussi être décrite manuellement par les experts du domaine. Les tables de probabilités sont calculées à partir de données expérimentales [COWELL, 1999].

2. Les réseaux bayésiens s'appuient sur le théorème de Bayes (1763)

3. ne contient pas de boucle

Les réseaux bayésiens sont utilisés dans plusieurs applications industrielles pour effectuer de la surveillance, à titre d'exemple. Connaissant la panne, un système basé sur des réseaux bayésiens peut déterminer les causes les plus probables ayant entraînées le défaut [MEDJAHHER, 2008]. Les réseaux bayésiens sont aussi utilisés dans l'analyse de risques, le datamining, la modélisation et la simulation de systèmes complexes comme par exemple l'analyse des courbes de santé [MARSHALL, 2002], [LE, 2011]. Les bases de données de l'assurance maladie contiennent la trace précise des courbes de soin des personnes. Il est alors possible de modéliser le comportement des personnes de façon probabiliste pour évaluer l'impact de l'évolution de la demande de soin sur les organisations existantes, notamment en ce qui concerne les personnes âgées.

2.4.3 Réseau de Petri

Les réseaux de Petri ont été introduits par [PETRI, 1962]. Ils servent à modéliser et à analyser le comportement dynamique de systèmes à événements discrets [MURATAT, 1989] évoluant d'un état à un autre à la suite d'événements internes ou externes. Les réseaux de Petri offrent une représentation graphique simple des systèmes modélisés.

Un réseau de Petri (FIGURE 2.5) est constitué des composants graphiques suivants :

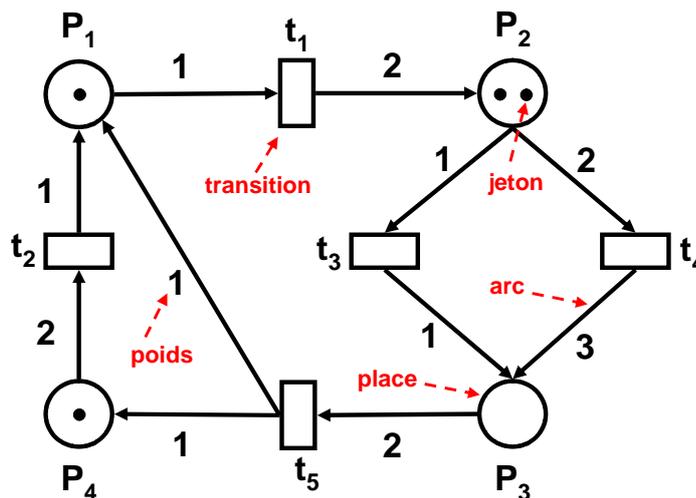


FIGURE 2.5 – Réseau de Petri

- les places représentées par des cercles, décrivent les différents états du système,
- les jetons représentés par des points, décrivent le nombre d'objets ou de ressources disponibles,
- les transitions représentées par des traits ou des rectangles, décrivent les évènements qui modifient les états du système,
- les arcs représentés par des flèches unidirectionnelles, relient les places aux transitions, et vice versa. Deux places (respectivement deux transitions) ne peuvent pas être reliées entre elles,
- les poids sont affectés aux arcs pour identifier le nombre de jetons nécessaires pour valider une transition.

L'état global d'un réseau de Petri se définit par un ensemble de places dont certaines sont marquées avec un ou plusieurs jetons et d'autres sont non marquées. La distribution des jetons dans les places à un instant donné est appelé marquage du réseau de Petri [KORDIK, 2008].

L'évolution d'un réseau de Petri est donc représentée par l'évolution de ses marquages. Si à un moment donné, le marquage du réseau ne permet aucune transition on dit que le système est bloqué. Dans les autres cas on parle d'un réseau vivant ou d'un réseau sans blocage [DAVID, 1997], [DIAZ, 2001].

Les premiers réseaux de Petri sont des modèles ordinaires ou généralisés⁴, très limités pour décrire des applications informatiques ou plus généralement industrielles. C'est pourquoi, des extensions des réseaux de Petri ont été introduites. Parmi elles, nous trouvons : Les réseaux de Petri temporisés [RAMCHANDANI, 1974], les réseaux à Prédicats/Transitions [GENRICH, 1981], les réseaux réguliers [HADDAD, 1987], les réseaux ordonnés [COURVEUR, 1988], les réseaux colorés [JENSEN, 1991], les réseaux algébriques [REISIG, 1991]

2.4.4 Digraphe

Le digraphe (FIGURE 2.6) est un graphe noté $G(S, A)$, déduit des équations d'état d'un système. Il est composé d'un ensemble de sommets S , où $S = \{U, Y, X\}$ correspondant aux entrées, sorties et états du système. Les interactions entre les nœuds sont représentées par des arcs orientés (A). Le graphe est appelé graphe signé si l'arc considère l'influence qualitative entre les variables [DION, 2003], [MAURYA, 2004].

4. Les arcs sont le plus souvent valués.

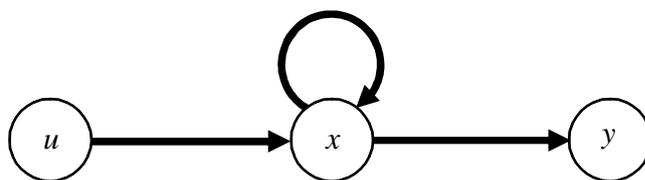


FIGURE 2.6 – Digraphe

2.4.5 Graphe fonctionnel

Le graphe fonctionnel (FIGURE 2.7) décrit un système à partir des services rendus par ses composants. Les services sont organisés en sous-ensembles cohérents, par rapport à une situation donnée et un ensemble d'objectifs à réaliser, correspondant à des modes d'utilisation. Un exemple de graphe fonctionnel est le graphe de gestion des modes d'utilisation qui spécifie les conditions de changement de modes en fonction de la disponibilité des services sous-jacents à chaque mode et des souhaits de l'utilisateur [GEHIN, 2008].

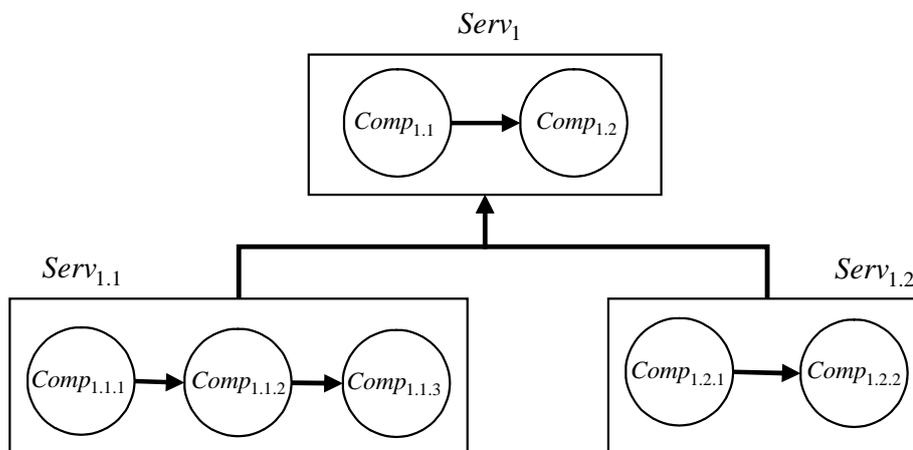


FIGURE 2.7 – Graphe fonctionnel

2.4.6 Graphe biparti

Un graphe est biparti si l'ensemble des sommets S peut être partitionné en deux sous-ensembles disjoints $S(Z, C)$. Z est l'ensemble de variables qui définit le comportement dynamique du système, et C est l'ensemble d'équations qui définit les relations entre les variables (FIGURE 2.8). Les relations entre

ces deux sous-ensembles sont représentées par des arcs A [BLANKE, 2006]. Le graphe biparti est beaucoup plus générique que le digraphe puisque l'en-

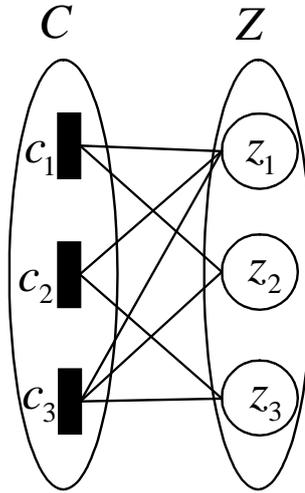


FIGURE 2.8 – Graphe biparti

semble d'équations C peuvent être des relations analytiques, floues, quantitatives, ou des tables.

2.4.7 Bond graph

Un bond graph (FIGURE 2.9-a) qui est aussi un graphe $G(S, A)$ est un outil de modélisation basé sur l'analogie et l'échange de puissance entre les systèmes. Il est bien adapté pour les systèmes multi-physiques puisque les variables de puissance, effort (e) et flux (f) sont génériques et indépendantes de la nature physique du système. Contrairement aux graphes précédents, S regroupe des composants matériels, des sous-systèmes, et d'autres éléments de base appelés jonctions, les arcs A , expriment l'échange de puissance entre les sous-systèmes. Chaque arc est représenté par une demi-flèche dont l'orientation indique le sens du transfert de la puissance. Cette dernière est décrite par deux variables de puissance conjuguées, à savoir l'effort e et le flux f .

Le modèle bond graph possède aussi des propriétés structurelles et causales indiquées par la position du trait causal sur le lien. Ainsi, sur la FIGURE 2.9-b le trait causal indique que le système 1 impose l'effort au Système 2 : $e = \phi(f)$, alors que sur la FIGURE 2.9-c nous avons $f = \phi(e)$. Ces propriétés sont exploitées pour l'analyse, l'observabilité, la commandabilité, ou la surveillabilité des systèmes [SAMANTARAY, 2008], [SUEUR, 1989].

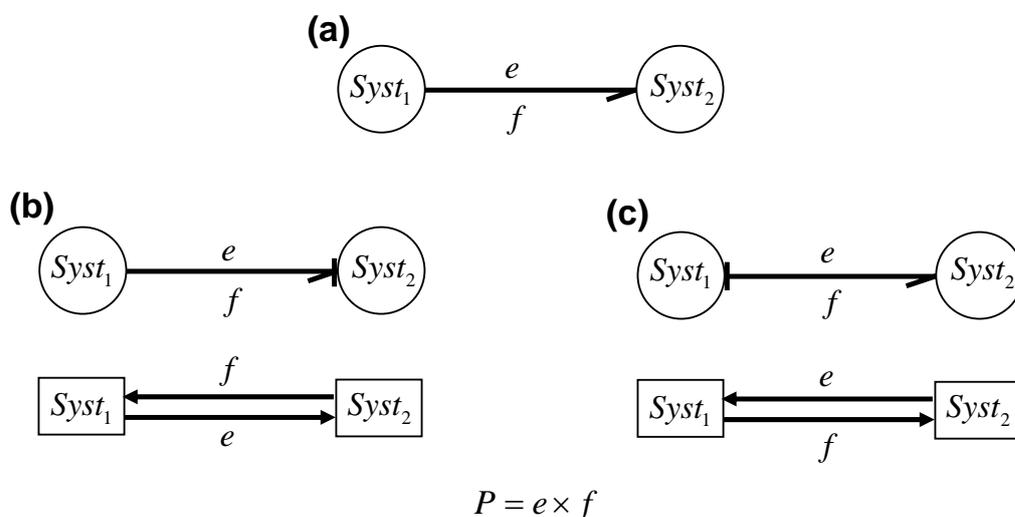


FIGURE 2.9 – Bond graph

2.4.8 Graphe

L'histoire de la théorie des graphes débute avec les travaux d'Euler au XVIII^e siècle (1736). L'article des ponts de Königsberg [EULER, 1736] est considéré comme le premier article abordant cette théorie. Il pose le problème suivant : *Est-il possible, en partant d'un quartier quelconque de la ville Königsberg, de traverser les sept ponts sans passer deux fois par le même et de revenir au point de départ ?* (FIGURE 2.10 - a).

D'autres problèmes ont une importance historique pour la théorie des graphes [CARTIER, 2008], nous citons :

- le problème du cavalier (FIGURE 2.10 - b) proposé par Euler en 1759 : *Un cavalier posé sur une case quelconque d'un échiquier doit en visiter toutes les cases sans passer deux fois sur la même,*
- le problème de coloriage de cartes (FIGURE 2.10 - c) énoncé en 1852 par Francis Guthrie : Il s'agit de colorier, en n'utilisant que quatre couleurs différentes, n'importe quelle carte découpée en régions connexes, de sorte que deux régions ayant une frontière en commun soient coloriées avec des couleurs différentes,
- le problème hamiltonien (FIGURE 2.10-d) de William Rowan Hamilton au XIX^e siècle : *Existe-t-il un chemin ou un circuit passant par tous les sommets d'un graphe une et une seule fois ?*

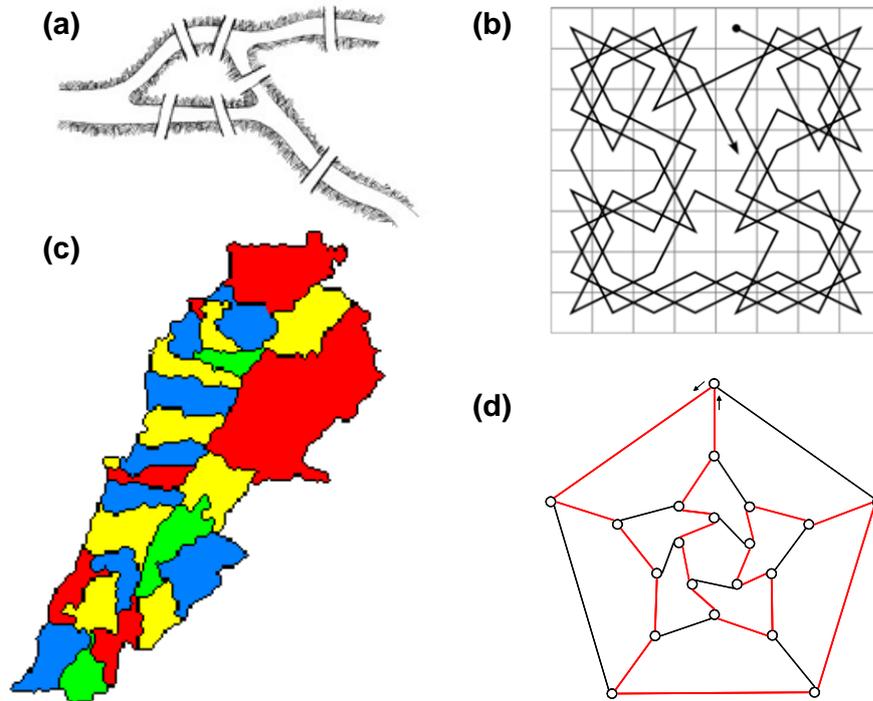


FIGURE 2.10 – (a) Ponts de Königsberg, (b) Problème du cavalier, (c) Problème de coloriage de cartes, (d) Problème hamiltonien.

De manière générale, un graphe permet de représenter la structure et les connexions entre éléments d'un ensemble complexe sous forme de relations. Un graphe est représenté par un ensemble fini non vide de sommets, reliés entre eux par des arcs ou arêtes. Quand on parle de graphe orienté, les sommets sont reliés par des arcs, sinon ils le sont par des arêtes. Un graphe non-orienté peut être vu comme un graphe orienté où chaque arête est composée de deux arcs, un dans chaque sens. Deux sommets reliés entre eux sont dits adjacents et un arc relié à un sommet donné est dit incident à ce sommet. Le degré d'un sommet est le nombre des arcs incidents à ce sommet. Pour un graphe orienté, le degré d'un sommet est la somme des arcs entrants et sortants de ce sommet. Un sommet de degré nul est appelé sommet isolé.

Graphe non-orienté

Un graphe non-orienté $G(S, A)$ est une relation binaire symétrique, où $S = \{S_i / i \in \mathbb{N}^*\}$ est un ensemble de cardinalité finie. Les éléments S_i sont appe-

lés sommets de G . $A = \{A_{i,j} = \{S_i, S_j\} / S_i, S_j \in S\} \subseteq S \times S$ est l'ensemble des arêtes de G . Chaque arête $A_{i,j}$ relie les deux sommets S_i et S_j entre eux (FIGURE 2.11).

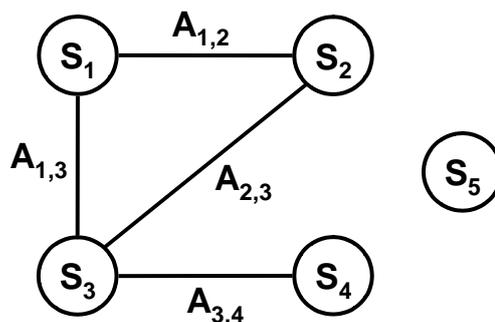


FIGURE 2.11 – Graphe non-orienté composé de cinq sommets et quatre arêtes.

Notons que parce qu'une arête dans un graphe non-orienté est un ensemble $\{S_1, S_2\} = \{S_2, S_1\}$, on ne peut pas avoir plus d'une arête entre deux sommets. Une autre conséquence est qu'on ne peut pas avoir une arête entre un sommet et lui-même dans un graphe non-orienté parce qu'une arête est un ensemble de deux éléments distincts.

Graphe orienté

Un graphe orienté $G(S, A)$ est une relation binaire, où $S = \{S_i / i \in \mathbb{N}^*\}$ est un ensemble de cardinalité finie dont les éléments sont appelés sommets de G et $A = \{A_{i,j} = (S_i, S_j) / S_i, S_j \in S\} \subseteq S \times S$ est l'ensemble des arcs de G . Chaque arc $A_{i,j}$ relie les deux sommets S_i et S_j entre eux, S_i est l'origine de l'arc, S_j est l'extrémité (FIGURE 2.12). Si $S_i = S_j$ on dit que $A_{i,j}$ est une boucle.

Un chemin est une suite de sommets connectés par des arcs, et un circuit est un chemin dont l'origine et l'extrémité coïncident. Un graphe non-orienté est un cas particulier de graphe orienté où la relation binaire est symétrique (FIGURE 2.13).

Graphe valué

Soit le graphe $G(S, A)$ et w une fonction qui pour chaque arc (ou arête) associe une information. Le graphe $G(S, A, w)$ est appelé un graphe valué.

Les graphes valués servent à résoudre de nombreux problèmes d'optimisation et d'ordonnancement (FIG 2.14).

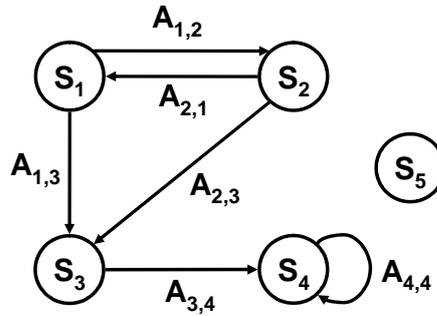


FIGURE 2.12 – Graphe orienté composé de cinq sommets et six arcs.

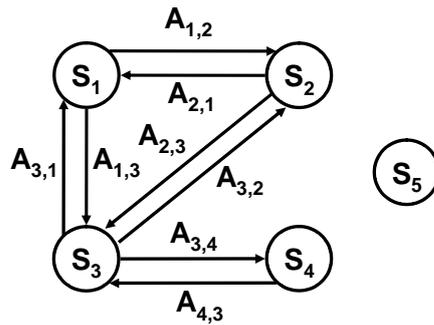


FIGURE 2.13 – Représentation du graphe non-orienté de la FIGURE 2.11 par un graphe orienté.

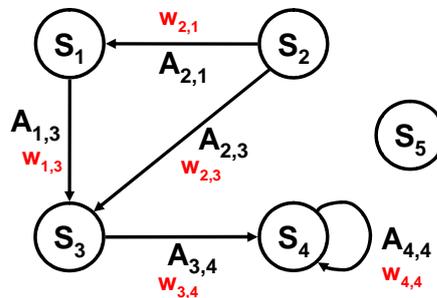


FIGURE 2.14 – Graphe orienté valué.

Sous-graphe

Un sous-graphe G' d'un graphe G est un graphe dont l'ensemble des sommets S' et des arcs (arêtes) A' sont des sous-ensembles de S et A respectivement. Comme chaque ensemble est un sous-ensemble de lui-même, chaque graphe est un sous-graphe de lui-même. Les sommets (nœuds) et arcs de G ne sont pas forcément présents dans G' , mais si un sommet est présent dans G' , un sommet correspondant est défini dans G , et toute arête qui relie deux sommets de G' doit également connecter les mêmes sommets dans G .

Formellement, un graphe $G'(S', A')$ est un sous-graphe du graphe $G(S, A)$ si $S' \subseteq S$, $A' \subseteq A$, et $\forall A'_{i,j} \in A' \Rightarrow S'_i, S'_j \in S'$ (FIGURE 2.15).

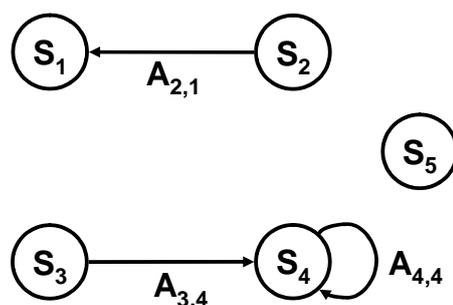


FIGURE 2.15 – Un sous-graphe du graphe orienté de la FIGURE 2.12.

La théorie des graphes s'applique à diverses disciplines telles que : infrastructures (réseaux de transports), gestion de projets (diagramme Gantt), informatique (topologie des réseaux), logistique (trafic routier), chimie (réactions), biologie (réseaux d'interactions protéiques), sociologie (réseaux d'affinités entre personnes), géographie (carte), productique (ordonnancement) En fonction de leur capacité à modéliser des situations qui peuvent évoluer dans le temps, on distingue, dans la littérature, les graphes dits « statique » des graphes dits « dynamiques » (FIGURE 2.16).

Graphe statique

Les graphes statiques sont utilisés en particulier pour visualiser des systèmes et des situations complexes de façon statique indépendamment du temps.

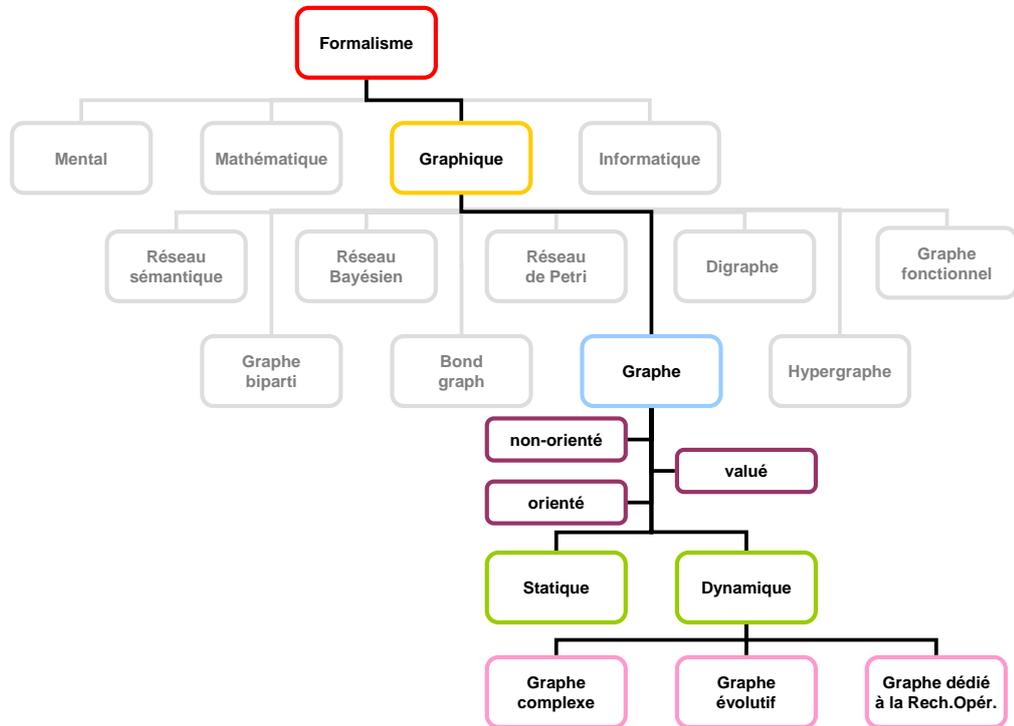


FIGURE 2.16 – Le graphe et ses dérivés

Graphe dynamique

Par rapport au graphe statique, un graphe dynamique peut modéliser l'interaction entre les composants « système » d'un ensemble de SdS. Un graphe dynamique peut aussi évoluer dans le temps, suivant l'état du composant « système » et son influence sur les autres composants.

Dans la littérature, il n'existe aucun formalisme générique pour décrire les graphes dynamiques et définir l'ensemble des éléments qui les caractérisent à un instant donné. Comme exemple de graphes dynamiques, on peut citer les graphes complexes, les graphes évolutifs, et les graphes dédiés à la recherche opérationnelle.

Graphe complexe

Ce type de graphe, appelé aussi réseau complexe, concerne les systèmes formés d'un grand nombre d'entités dynamiques fortement interconnectées [BOCCALETTI, 2006]. L'exemple le plus connu dans ce type de réseau est le Web, qui évolue en permanence, où des millions de nouvelles pages sont

créées, mises à jour, ou supprimées. Cet ensemble de composants (pages) peut être représenté par un graphe dynamique, appelé graphe du Web, dont les sommets correspondent aux pages Web et les arcs aux hyperliens. Chaque sommet de cet énorme graphe est identifié par un URL ⁵.

Le graphe du Web est un graphe dynamique où des pages et des liens apparaissent, disparaissent et changent à chaque instant. [DEO, 2007] a proposé un mode de construction en tenant compte des modifications dynamiques du Web. Ce mode contient des mécanismes d'ajout et de suppression de sommets et d'arcs, où le graphe est construit de manière itérative, étape par étape. A chaque étape, deux possibilités sont offertes, soit un sommet est ajouté avec un lien sur le graphe actuel, soit un sommet est retiré avec ses arcs incidents.

Graphe évolutif

Les graphes évolutifs permettent de représenter des interactions qui évoluent de manière discrète dans le temps. Ces graphes ont été utilisés comme un modèle de réseau dynamique dans plusieurs travaux. Dans [BUI-XUAN, 2003] et [FERREIRA, 2002] la définition du graphe évolutif a été introduite de la façon suivante :

Soient $G(S, A)$ un graphe et $SG = G_1, G_2, \dots, G_T$ une séquence ordonnée de ses sous-graphes, avec $\bigcup_{i=1}^T G_i = G$. Le système $GE = (G, SG)$ est appelé graphe évolutif (FIGURE 2.17).

Dans le cas d'un SdS, un ensemble de composants systèmes peuvent évoluer dans le temps en formant un graphe évolutif GE , qui représente l'union de tous les sommets et les arcs (arêtes) durant la vie du SdS. Chaque sous-graphe G_i représente l'état du SdS à une étape donnée, où chaque arc (arête) est étiqueté par un indice précisant une étape d'évolution (FIGURE 2.17). Pour chaque arc (arête) de ce graphe évolutif, on peut associer un poids correspondant au temps nécessaire pour échanger l'information entre deux composants « système ».

Graphe dédié à la recherche opérationnelle

La recherche opérationnelle est un ensemble de méthodes d'analyse (mathématique et informatique) des phénomènes d'organisation qui traite de la

5. Le sigle URL (Uniform Resource Locator), auquel se substitue informellement le terme adresse web, désigne une chaîne de caractères utilisée pour adresser les ressources du World Wide Web : document HTML, image, son, boîte aux lettres électronique ...

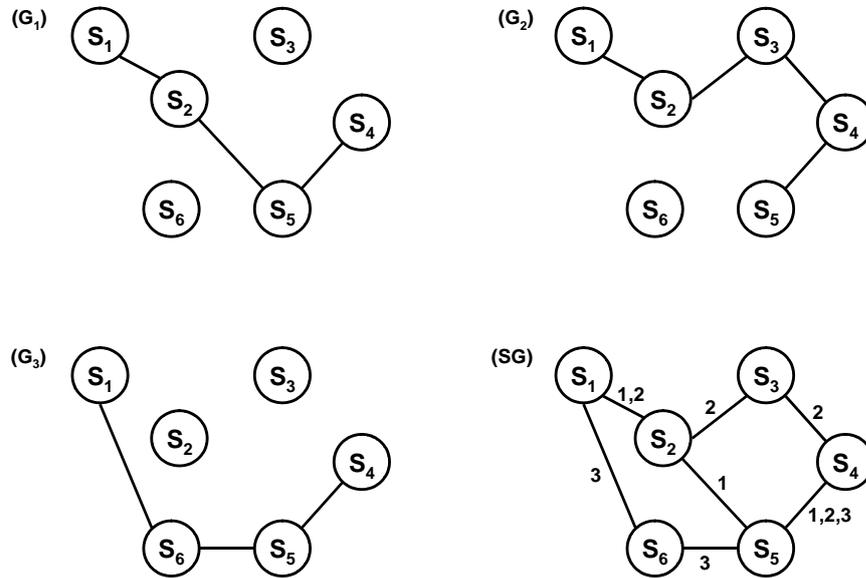


FIGURE 2.17 – Exemple d'un graphe évolutif dynamique (SG) construit à partir des trois graphes statiques (G_1 , G_2 et G_3) représentant des graphes évolutifs.

maximisation d'un profit, d'une performance, d'un rendement ou bien de la minimisation d'un coût, d'une dépense. La recherche opérationnelle est avant tout un outil d'aide à la décision. Elle est également liée à l'optimisation combinatoire, l'algorithmique et la théorie des graphes.

Souvent, les problématiques liées à la recherche opérationnelle peuvent être résolues par la théorie des graphes, comme par exemple les solutions d'optimisation dans les systèmes de production. Les techniques d'optimisation en ligne s'inscrivent dans le contexte des graphes dynamiques. Parmi les travaux qui traitent des graphes dynamiques pour la recherche opérationnelle, nous citons ceux de [RAMALINGAM, 1996] et [DEMETRESCU, 2006] qui mettent à jour la structure d'un graphe statique $G(S, A)$ lors de l'arrivée d'événements nouveaux (ajout, suppression, ou modification d'un arc) afin de recalculer le plus court chemin. Ce graphe évolue de manière discrète, étape après étape. A chaque étape, un nouveau graphe statique est construit à partir du graphe existant, prenant ainsi en compte l'arrivée des nouveaux événements. Le graphe dynamique est alors construit de manière itérative à partir d'une suite discrétisée de graphes statiques.

2.4.9 Hypergraphe

Les hypergraphes généralisent le concept du graphe en introduisant la notion d'hyperarête. Dans ce cas, l'arête du graphe n'est pas soumise à une restriction sur le nombre de sommets connectés. Autrement dit, les arêtes ne relient plus un ou deux sommets, mais un nombre quelconque de sommets [BERGE, 1987].

Un hypergraphe H est un couple (S, E) tel que $S = \{S_1, S_2, \dots, S_n\}$ est un ensemble fini non vide de sommets de H et $E = \{E_1, E_2, \dots, E_m\} \subseteq \binom{S}{1..n}$ est une famille de parties non vides de S représentant les hyperarêtes de H , avec $\bigcup_{i=1}^m E_i = S$. On note $\binom{S}{1..n}$ l'ensemble des parties non vides de S contenant au moins un élément et au plus n éléments (FIGURE 2.18). Remarquons qu'avec cette définition, une arête peut contenir tous les sommets de l'hypergraphe, sachant qu'il n'existe pas de sommet isolé.

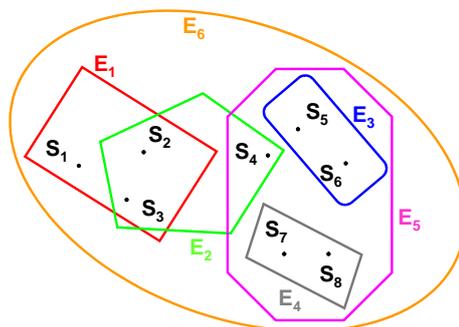


FIGURE 2.18 – Hypergraphe composé de huit sommets et six hyperarêtes.

Ainsi, le graphe est une entité de l'hypergraphe. En effet, le graphe est un cas particulier de l'hypergraphe, où chaque élément de l'ensemble S a exactement deux éléments. Autrement dit, un graphe est un hypergraphe avec $E \subseteq \binom{S}{2..2}$ (FIGURE 2.19).

Par rapport aux propriétés graphiques de l'hypergraphe qui sont développées dans le chapitre 3, nous allons démontrer que ce formalisme graphique est mieux adapté à la modélisation multi-niveaux et multi-structures d'un SdS, pour la synthèse d'une architecture globale de supervision.

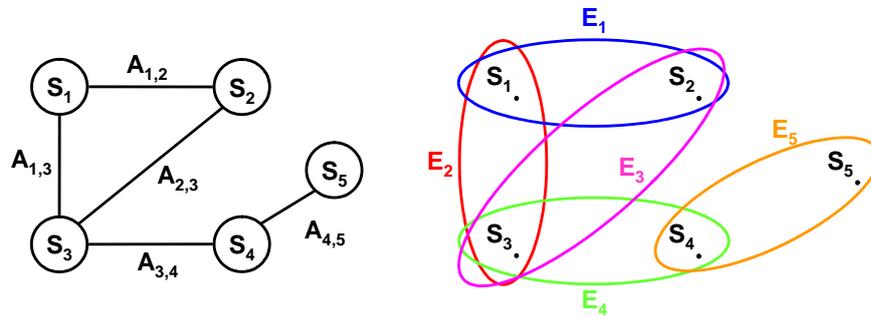


FIGURE 2.19 – (a) graphe et (b) hypergraphe.

2.5 Conclusion

Dans ce chapitre, nous avons positionné notre travail de recherche doctorale par rapport à la problématique liée à la modélisation des SdS. Pour cela, nous avons décrit les caractéristiques fondamentales d'un SdS afin d'introduire une définition appropriée à ce concept. Le choix d'un formalisme de modélisation graphique dans notre développement se justifie par :

- la généralité, où à travers le même modèle graphique, nous pouvons faire de l'analyse, de l'optimisation et prendre une décision suivant l'évolution de l'état des composants systèmes,
- la simplicité en l'absence de calcul et l'indépendance aux paramètres des composants systèmes.

Notre contribution scientifique, à travers ce travail de recherche, concerne :

1. **Développement de modèles graphiques d'un SdS** : les modèles graphiques d'un SdS sont représentés sur plusieurs niveaux et respectant ses propriétés fondamentales à savoir : l'indépendance opérationnelle et managériale, la coopération, la dispersion et le développement évolutionnaire. Deux modèles graphiques ont été proposés pour deux objectifs distincts : la supervision et l'optimisation.
2. **Exploitation du modèle SdS pour la supervision** : le premier est basé sur l'approche de l'hypergraphe, où la notion de l'hyperarête est introduite afin de généraliser la liaison graphique (arête) entre deux nœuds par une liaison multi-nœuds et multi-niveaux. Il a été démontré dans ce travail de recherche, qu'à travers le modèle de l'hypergraphe, les propriétés d'un SdS sont préservées. Une décomposition hiérarchique

d'un SdS sous forme d'hypergraphes va nous permettre d'élaborer une stratégie de supervision. En fait, le parcours ascendant de l'hypergraphe permet de suivre les conséquences d'un changement de mode de fonctionnement d'un composant système, à un niveau hiérarchique donné, sur l'ensemble des composants systèmes de niveau supérieur et ainsi de détecter le mode de fonctionnement global du SdS. A travers le parcours descendant, une optimisation verticale (multi-niveaux) est réalisée afin de définir des reconfigurations sur le SdS, permettant de maintenir la réalisation de sa mission globale avec des performances identiques ou inférieures.

3. **Exploitation du modèle SdS pour l'optimisation** : le deuxième modèle graphique de SdS est dérivé du modèle de l'hypergraphe mais se focalise sur un niveau particulier de modélisation. Ce modèle dynamique et évolutif dépendant du temps, permet d'élaborer des algorithmes d'optimisation en ligne pour les composants systèmes associés à un niveau de modélisation. Pour ce type d'optimisation dite transversale, un modèle multi-niveaux par hypergraphe n'est pas adéquat. En effet, les interactions entre ce niveau de modélisation et les autres niveaux ont besoin d'être représentées par des poids de fonctions agissant directement sur les nœuds et les arcs. Ce modèle a été exploité pour le calcul du temps optimal et adaptatif, qu'un véhicule intelligent autonome (IAV) prend pour relier un nœud de départ et un nœud de destination dans un réseau évolutif d'un système ITS.

CHAPITRE 3

Modélisation des systèmes de systèmes par l'approche de l'hypergraphe

Sommaire

3.1	Introduction	50
3.2	Représentation multi-niveaux et multi-structures d'un système de systèmes	51
3.3	Modélisation d'un système de systèmes par l'approche de l'hypergraphe	53
3.4	Propriétés du système de systèmes à partir du modèle de l'hypergraphe	59
3.5	Supervision d'un système de systèmes à base d'un modèle hypergraphe	64
3.5.1	Architecture du système de supervision	64
3.5.2	Surveillance	65
3.5.3	Gestion des modes de fonctionnement sur l'hypergraphe	66
3.5.4	Parcours descendant de l'hypergraphe pour la reconfiguration	71
3.6	Application à un système de transport intelligent	74
3.6.1	Modélisation du système de transport intelligent	76
3.6.2	Surveillance	79
3.6.3	Identification des modes de fonctionnement	80

3.6.4 Reconfiguration	84
3.7 Conclusion	85

3.1 Introduction

La modélisation est devenue un outil incontournable pour étudier, analyser voir prédire la dynamique des systèmes de plus en plus complexes. Le formalisme mathématique a été depuis longtemps utilisé pour décrire les relations entre les différents paramètres du système modélisé. Cette approche nécessite souvent la connaissance parfaite du système ainsi que de ses paramètres, afin de représenter au mieux son comportement par des formulations mathématiques.

Par ailleurs, le formalisme graphique semble être prometteur au vue de sa capacité à représenter facilement la complexité du système à modéliser, sans avoir une connaissance *a priori* sur ses paramètres, en absence de calculs complexes préalables.

Le formalisme graphique est bien positionné pour décrire l'interaction entre les sous systèmes du système global, mais aussi entre les composants élémentaires de chaque sous système. Il consiste à représenter chaque composant système par un sous-graphe, évoluant dynamiquement par rapport aux objectifs ou l'état du composant système.

Un Système de Systèmes (SdS) est constitué de composants systèmes qui sont répartis sur plusieurs niveaux hiérarchisés : micro, méso et macroscopique, où pour chaque niveau, un composant système peut avoir plusieurs structures. Un tel système nécessite un modèle graphique représentatif du composant système pour chaque niveau et pour chaque structure. Ce modèle doit représenter le composant système par rapport à son indépendance opérationnelle, son interaction avec les autres composants systèmes, sa dispersion géographique, ainsi que ses évolutions dynamique et temporelle.

Nous avons présenté dans le chapitre 2 plusieurs formalismes graphiques de modélisation. Rares ceux qui arrivent à décrire précisément le SdS, tel qu'il est défini avec ses caractéristiques fondamentales. Par contre, les hypergraphes peuvent modéliser l'interaction entre les composants d'un SdS grâce à son architecture graphique.

Ainsi, dans ce chapitre, nous proposons d'étudier les points suivants :

- la représentation multi-niveaux et multi-structures d'un SdS ;
- la modélisation par l'approche de l'hypergraphe d'un SdS ;
- la supervision d'un SdS à partir du modèle hypergraphe ;
- l'application d'un SdS au système de transport intelligent.

3.2 Représentation multi-niveaux et multi-structures d'un système de systèmes

De façon générale, la modélisation des SdS peut être vue selon plusieurs niveaux suivant les objectifs définis, où chaque niveau peut décrire plusieurs structures (FIGURE 3.1).

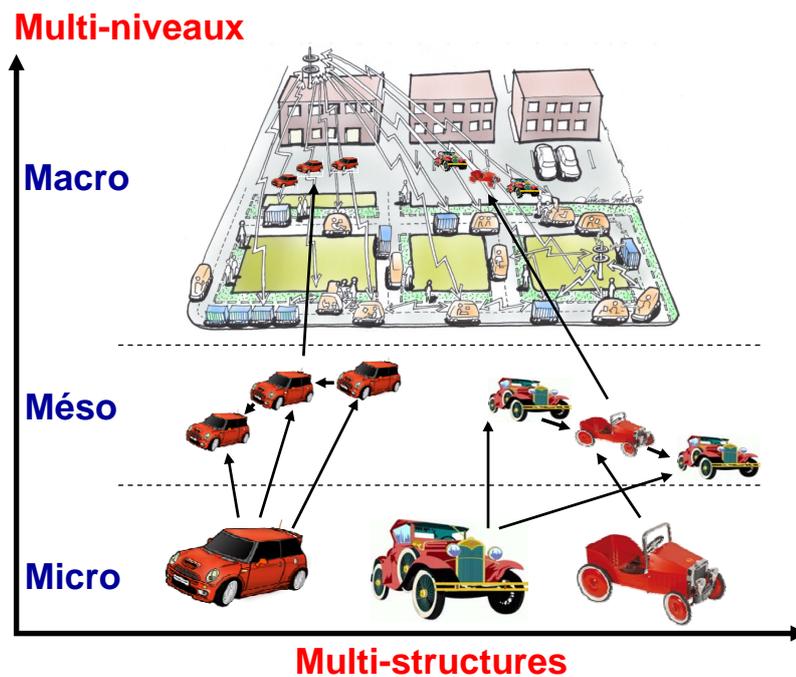


FIGURE 3.1 – Exemple d'un modèle multi-niveaux et multi-structures appliqué au transport

Lorsque nous prenons l'exemple d'un véhicule routier, à un niveau bas dit microscopique, nous nous intéressons au comportement dynamique des différentes parties du véhicule. Pour cela, la connaissance parfaite des paramètres du véhicule nous permet d'élaborer un modèle dynamique plus fin et mieux

représentatif. Ce modèle peut être exploité pour élaborer des algorithmes de type contrôle/commande ou surveillance. À ce niveau, sont utilisés des modèles de comportement du système physique pour son pilotage. Le véhicule est alors considéré comme une entité indépendante de l'environnement. Le modèle informationnel n'est pas exploité. À un niveau supérieur dit mésoscopique, nous nous intéressons plutôt à une flotte de véhicules avec des structures homogènes ou hétérogènes. A ce niveau, nous pouvons voir le système « véhicule » comme un objet cinématique se déplaçant dans l'espace et coopérant ou coexistant avec d'autres véhicules. Un modèle décrivant la connectivité du système d'information véhicule à véhicule a toute son importance.

Ainsi, le modèle représentatif d'un tel niveau peut décrire la densité du flux de véhicules en fonction de la vitesse moyenne, sans avoir besoin de connaître d'autres détails sur les véhicules. Ce modèle peut servir à l'élaboration d'algorithmes pour la gestion de trafic. En l'absence d'interaction entre les niveaux microscopique et mésoscopique, les composants systèmes sont considérés souvent en fonctionnement normal (non défaillant). Lorsque nous souhaitons modéliser un trafic de véhicules dans un environnement urbain, en considérant l'infrastructure et d'autres éléments de l'environnement, nous passons à un niveau macroscopique. A ce stade de modélisation, nous nous intéressons souvent à l'interaction entre le véhicule et l'infrastructure, le positionnement géodynamique, la sécurité ... en présence de la ICT. Ainsi, d'autres niveaux peuvent être définis selon la finesse du modèle désiré et en fonction des objectifs de la modélisation.

Un modèle multi-niveaux et multi-structures d'un SdS (FIGURE 3.2) est constitué d'un ensemble de composants systèmes notées $CS_{i,j}^k$. Chaque $CS_{i,j}^k$ représente le $i^{\text{ème}}$ système composant appartenant au $j^{\text{ème}}$ niveau d'abstraction et ayant la $k^{\text{ème}}$ structure. Pour chaque composant système de niveau j , on peut dériver d'autres composants systèmes dits de raffinement ($CS_{i,j-1}^k, \dots, CS_{i',j'}^{k'}$) de niveaux inférieurs [SANCHO, 2001].

En général, les composants systèmes de niveaux supérieurs sont choisis suivant les missions définies. Les composants systèmes de niveaux inférieurs évoluent en fonction des changements de missions des composants systèmes de niveaux supérieurs.

La problématique de modélisation pour ce type d'architecture réside dans le choix d'un formalisme permettant de représenter l'interaction entre les différents niveaux. Ainsi, un modèle macroscopique peut exploiter les résul-

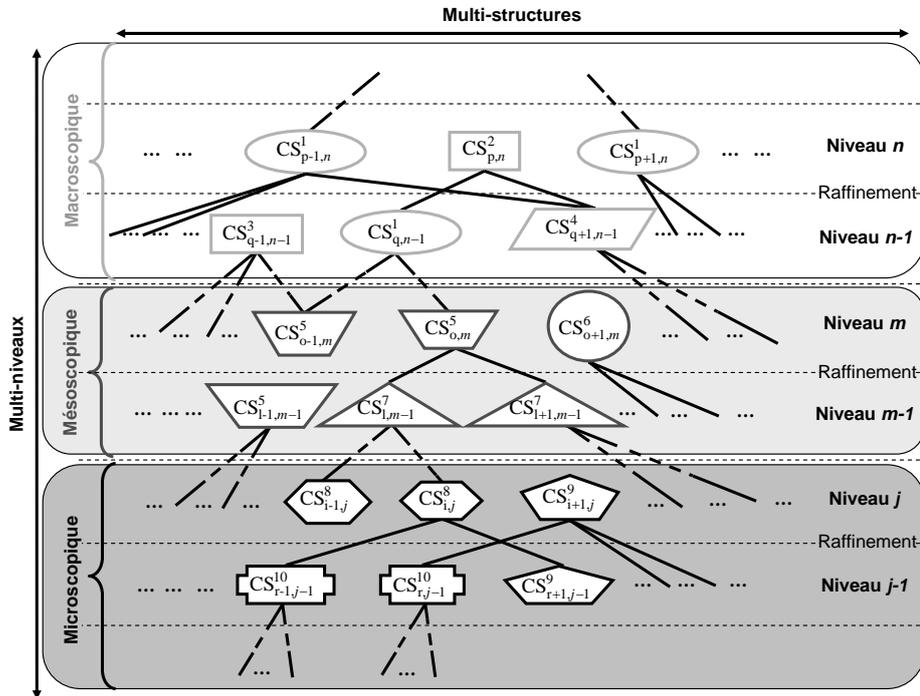


FIGURE 3.2 – Modèle multi-niveaux et multi-structures

tats obtenus pour le modèle mésoscopique de niveau inférieur et ainsi de suite.

Un SdS, compte tenu de ses propriétés fondamentales, peut être représenté verticalement sur plusieurs niveaux, suivant la gestion managériale ou opérationnelle du composant système. De même il peut être représenté transversalement suivant l'hétérogénéité et la dispersion des composants systèmes. Dans les sections suivantes, nous allons montrer que le formalisme graphique des hypergraphes permet de modéliser l'interaction entre niveaux d'un SdS.

3.3 Modélisation d'un système de systèmes par l'approche de l'hypergraphe

Avant d'introduire le principe de l'hypergraphe comme une extension du graphe, il est important de définir la notion d'hyperensemble comme une généralité des ensembles pour la représentation de systèmes multi-niveaux et multi-structures. Dans ce qui suit et pour des raisons de simplification de formalisme, nous considérons seulement le $i^{\text{ème}}$ élément de la $j^{\text{ème}}$ abstraction d'un composant système $CS_{i,j}$ sans prendre en considération sa structure.

Définition 3.1 : Un hyperensemble¹ \mathcal{HE} est un ensemble dont les éléments sont simples² ou hyperéléments³ [CHEMERO, 2008].

Soit l'exemple de l'hyperensemble de la FIGURE 3.3 :

$$\mathcal{HE} = \{ \{ \{ CS_{1,0}, CS_{2,0} \}, CS_{3,0} \}, \{ CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0} \}, \{ CS_{7,0} \} \},$$

où $CS_{1,0}, CS_{2,0}, CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0}, CS_{7,0}$ sont les composants systèmes élémentaires de niveau 0 qui sont supposés indivisibles en d'autres composants systèmes. Cet hyperensemble est composé de quatre éléments :

- trois éléments simples :
 - $CS_{1,1} = \{ CS_{1,0}, CS_{2,0} \},$
 - $CS_{2,1} = \{ CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0} \},$
 - $CS_{3,1} = \{ CS_{7,0} \},$
- et un hyperélément :
 - $CS_{1,2} = \{ \{ CS_{1,0}, CS_{2,0} \}, CS_{3,0} \}.$

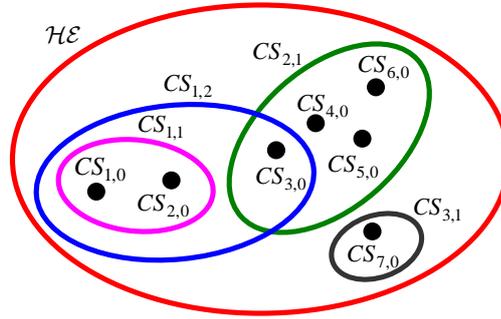


FIGURE 3.3 – Hyperensemble \mathcal{HE}

Définition 3.2 : On dit que $CS_{i,j}$ est un élément de \mathcal{HE} tel que $j \neq 0$, noté $CS_{i,j} \in \mathcal{HE}$, si et seulement si :

- $CS_{i,j} \in \mathcal{HE}$, ou
- $CS_{i,j} \in CS_{i',j'}$ avec $CS_{i',j'} \in \mathcal{HE}$.

On note $CS_{i,j} \in CS_{i',j'}$ un élément $CS_{i,j}$ qui appartient directement à un élément $CS_{i',j'}$ de niveau supérieur, et par $CS_{i,j} \in \mathcal{HE}$ un élément $CS_{i,j}$

1. dit aussi ensemble d'ensembles
 2. ensembles simples
 3. ensemble d'éléments

qui appartient à un hyperélément de $CS_{i',j'}$. Exemples :

$$\left. \begin{array}{l} CS_{1,1} \in CS_{1,2} \\ CS_{1,2} \in \mathcal{HE} \end{array} \right\} \Rightarrow CS_{1,1} \in \mathcal{HE}.$$

Définition 3.3 : Un hyperensemble est dit imbriqué (FIGURE 3.4) si et seulement si pour tout $CS_{i,j}, CS_{i',j'} \in \mathcal{HE}$, une des propriétés suivantes est vérifiée :

1. indépendance

$$CS_{i,j} \cap CS_{i',j'} = \emptyset,$$

2. inclusion

$$CS_{i,j} \subseteq CS_{i',j'} \text{ ou}$$

$$CS_{i',j'} \subseteq CS_{i,j}.$$

L'hyperensemble de la FIGURE 3.3 n'est pas un hyperensemble imbriqué, du fait que : $CS_{1,2} \cap CS_{2,1} \neq \emptyset$, $CS_{1,2} \not\subseteq CS_{2,1}$ et $CS_{2,1} \not\subseteq CS_{1,2}$. Dans l'exemple de la FIGURE 3.4, nous remarquons que l'une des propriétés indépendance ou inclusion est vérifiée.

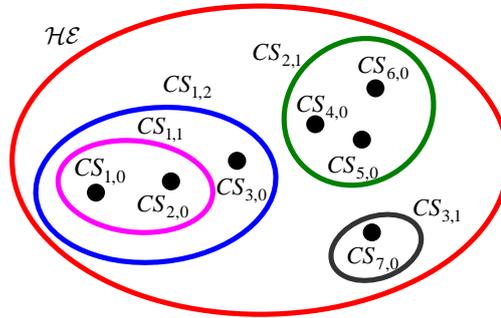


FIGURE 3.4 – Hyperensemble imbriqué

Par rapport à la théorie des hyperensembles, celle des hypergraphes se propose de généraliser la théorie des graphes en introduisant le concept d'hyperarête. Contrairement à une arête, une hyperarête n'impose pas de restriction sur le nombre de sommets (FIGURE 3.5). Ainsi, le graphe est un cas particulier de l'hypergraphe, où chaque élément de l'ensemble de sommets \mathcal{V} peut se lier au plus avec deux éléments. Autrement dit, un graphe est un hypergraphe où l'ensemble des hyperarêtes est borné par deux sommets, c'est à dire, $\mathcal{E} \subseteq \binom{\mathcal{V}}{1..2}$. On note $\binom{\mathcal{V}}{1..n}$ l'ensemble des parties non vides de \mathcal{V} contenant au moins un élément et au plus n éléments.

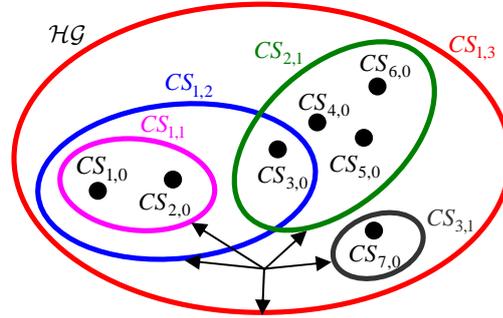


FIGURE 3.5 – Représentation ensembliste d'un hypergraphe composé de sept sommets et cinq hyperarêtes interconnectées.

Dans la représentation des systèmes par l'hyperensemble [CHEMERO, 2008], l'interconnexion entre les différents systèmes n'est pas prise en compte. Ainsi, cette représentation reste incomplète pour représenter un SdS et ses cinq propriétés fondamentales, d'où l'intérêt de modéliser un tel système par l'approche de l'hypergraphe.

Définition 3.4 : Formellement, un hypergraphe \mathcal{HG} est une paire $(\mathcal{V}, \mathcal{E})$, où $\mathcal{V} = \{CS_{i,0}, i \in \mathbb{N}\}$ est un ensemble fini de sommets au niveau d'abstraction 0, à partir duquel chaque $CS_{i,0}$ est élémentaire et ne peut pas se diviser en d'autres CS , et $\mathcal{E} = \{CS_{i,j}, i \in \mathbb{N}, j \in \mathbb{N}^*\}$ est un ensemble de sous-ensembles non-vides de \mathcal{V} liés entre eux, appelés hyperarêtes, avec $\mathcal{E} \subseteq 2^{\mathcal{V}}$, et $\bigcup_{i \in \mathbb{N}, j \in \mathbb{N}^*} CS_{i,j} = \mathcal{V}$.

Par comparaison avec l'hyperensemble, les sommets \mathcal{V} et les hyperarêtes \mathcal{E} de l'hypergraphe correspondent respectivement aux éléments simples $\{CS_{i,0}\}_{i \in \mathbb{N}}$ et aux hyperéléments interconnectés $\{CS_{i,j}\}_{i \in \mathbb{N}, j \in \mathbb{N}^*}$ de l'hyperensemble. Autrement dit, un hypergraphe est un hyperensemble dont les éléments (simples ou hyperéléments) sont liés entre eux par des relations (FIGURE 3.5).

Ainsi, une hyperarête peut contenir tous les sommets de l'hypergraphe, et un sommet doit appartenir au moins à une hyperarête [ARPINAR, 1999].

Définition 3.5 : Un hypergraphe imbriqué est un hypergraphe dont les sommets et les hyperarêtes forment un hyperensemble imbriqué (FIGURE 3.6), où les sous ensembles sont interconnectés.

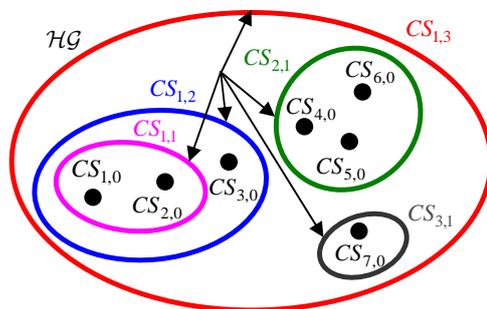


FIGURE 3.6 – Hypergraphe imbriqué

Définition 3.6 : Un hypergraphe $\mathcal{HG}^*(\mathcal{V}^*, \mathcal{E}^*)$ de la FIGURE 3.7 est dit dual de $\mathcal{HG}(\mathcal{V}, \mathcal{E})$ si et seulement si :

$$\begin{aligned}
 CS_{i,j} \in \mathcal{V}^* \text{ de } \mathcal{HG}^* &\iff CS_{i,j} \in \mathcal{E} \text{ de } \mathcal{HG} \\
 CS_{i,0} \in \mathcal{E}^* \text{ de } \mathcal{HG}^* &\iff CS_{i,0} \in \mathcal{V} \text{ de } \mathcal{HG}
 \end{aligned}$$

Sur cet hypergraphe dual, les liaisons entre les nouveaux sommets correspondent aux liaisons entre les hyperarêtes de l'hypergraphe initial. D'autres liaisons peuvent aussi apparaître entre les nouvelles hyperarêtes comme le montre la FIGURE 3.7.

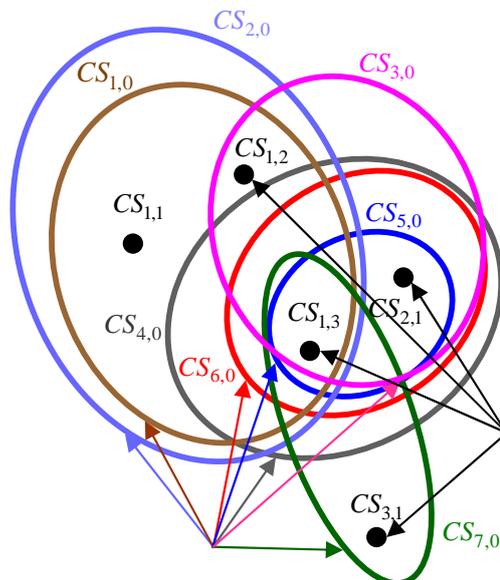


FIGURE 3.7 – Hypergraphe dual de celui de la FIGURE 3.5.

Définition 3.7 : Un hypergraphe $\mathcal{H}(\mathcal{V}, \mathcal{E})$ peut être représenté par le graphe valué $\mathcal{G}^v(\mathcal{V}^v, \mathcal{E}^v, \mathcal{F})$ semblable à celui présenté dans la FIGURE 3.8, où :

1. les sommets \mathcal{V}^v de \mathcal{G}^v représentent les hyperarêtes \mathcal{E} de \mathcal{H} :

$$\mathcal{V}^v = \{CS_{i,j}\}_{i \in \mathbb{N}, j \in \mathbb{N}^*},$$

2. deux sommets $CS_{i,j}$ et $CS_{i',j'}$ de \mathcal{G}^v sont liés entre eux si et seulement si, il existe une intersection non vide entre $CS_{i,j} \in \mathcal{E}$ et $CS_{i',j'} \in \mathcal{E}$:

$$\mathcal{E}^v = \{E_{i,j}^{i',j'} = (CS_{i,j}, CS_{i',j'}) / CS_{i,j} \cap CS_{i',j'} \neq \emptyset\},$$

3. chaque arête $E_{i,j}^{i',j'} \in \mathcal{E}$ de \mathcal{G}^v est valuée par la fonction \mathcal{F} définie comme suit :

$$\mathcal{F} : E_{i,j}^{i',j'} \mapsto \mathcal{F}_{i,j}^{i',j'} = CS_{i,j} \cap CS_{i',j'}.$$

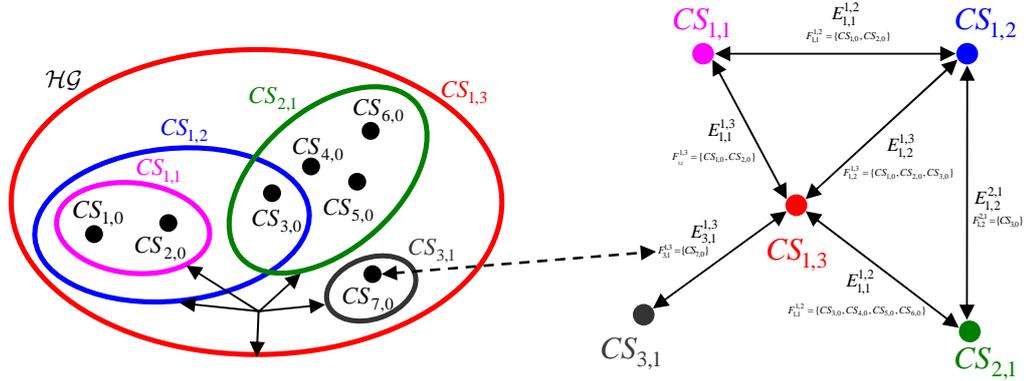


FIGURE 3.8 – Hypergraphe \mathcal{HG} , à gauche : représentation ensembliste, à droite : graphe valué.

Définition 3.8 : Un hypergraphe $\mathcal{HG}(\mathcal{V}, \mathcal{E})$ est dit valué (FIGURE 3.9) s'il est muni d'une application définie par :

$$\mathcal{M} : \mathcal{HG}(\mathcal{V}, \mathcal{E}) \mapsto \mathbb{T},$$

où \mathbb{T} est un domaine à définir selon le cas à étudier. Ainsi l'hypergraphe sera noté $\mathcal{HG}(\mathcal{V}, \mathcal{E}, \mathcal{M})$.

A partir de ces définitions, un hypergraphe $\mathcal{HG}(\mathcal{V}, \mathcal{E}, \mathcal{M})$ est considéré comme

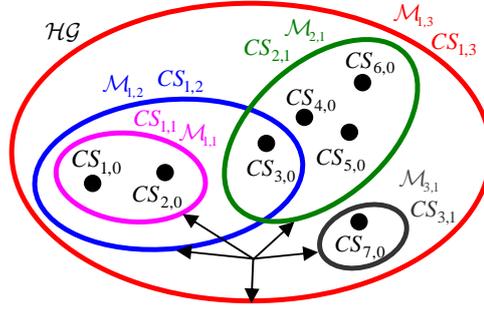


FIGURE 3.9 – Hypergraphe valué de la FIGURE 3.5.

un des meilleurs candidats à la modélisation graphique d'un SdS représenté sur plusieurs niveaux, dans le sens où les sommets \mathcal{V} de l'hypergraphe correspondent aux composants systèmes du plus bas niveau de modélisation, et où les hyperarêtes \mathcal{E} représentent l'ensemble de composants systèmes de niveau supérieur et leur interactions.

Un SdS dont lequel chaque composant système $CS_{i,j}/j \neq 0$ doit satisfaire un nombre fini $l_{i,j}$ de missions, peut être représenté par un hypergraphe imbriqué et valué. Les missions sont représentées sur l'hypergraphe par la fonction de valuation des hyperarêtes définie comme suit :

$$\begin{aligned} \mathcal{M} &: \mathcal{E} \mapsto \mathcal{M}(\mathcal{E}) \\ CS_{i,j} &\mapsto \mathcal{M}_{i,j} = \{M_{i,j}^{l_{i,j}}\} \end{aligned}$$

3.4 Propriétés du système de systèmes à partir du modèle de l'hypergraphe

Propriété 1 - indépendance opérationnelle : les composants systèmes $CS_{i',j'}$ d'un système $CS_{i,j}$ sont indépendants opérationnellement si et seulement si pour tout $CS_{i',j'} \in CS_{i,j}$:

$$CS_{i',j'} \in \mathcal{E}.$$

Dans ce cas, chaque composant système opère indépendamment des autres composants systèmes.

Le composant système $CS_{1,1}$ de la FIGURE 3.10 est indépendant opérationnellement, par contre le composant système $CS_{3,0}$ ne l'est pas, car il ne représente pas une hyperarête.

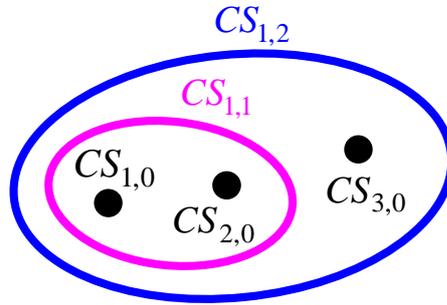


FIGURE 3.10 – Indépendance opérationnelle

Propriété 2 - indépendance managériale : les composants systèmes d'un système $CS_{i,j}$ sont indépendants managérialement si et seulement si pour tout $CS_{i',j'}, CS_{i'',j''} \in CS_{i,j}$:

$$\mathcal{M}_{i',j'} \cap \mathcal{M}_{i'',j''} = \emptyset.$$

Dans ce cas, chaque composant système gère ses propres missions indépendamment des autres composants systèmes.

Le composant système $CS_{3,1}$ de la FIGURE 3.11 est indépendant managérialement, à la différence des composants systèmes $CS_{1,1}$ et $CS_{2,1}$.

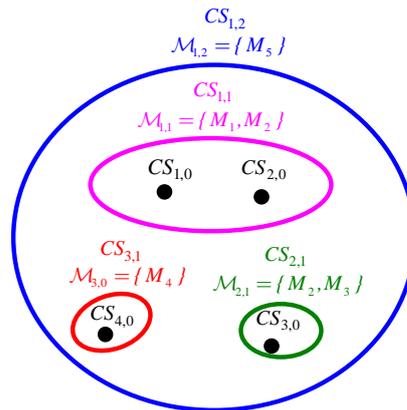


FIGURE 3.11 – Indépendance managériale

Propriété 3 - dispersion géographique : les composants systèmes $CS_{i',j'}$ et $CS_{i'',j''}$ d'un système $CS_{i,j}$ sont dispersés géographiquement si et seule-

ment si pour tout $CS_{i',j'}, CS_{i'',j''} \in CS_{i,j}$:

$$CS_{i',j'} \cap CS_{i'',j''} = \emptyset.$$

Dans ce cas, aucun lien physique entre les composants systèmes n'est autorisé pour réaliser les différentes missions, seul des échanges d'informations entre composants systèmes sont possibles.

Le composant système $CS_{3,1}$ de la FIGURE 3.12 est dispersé géographiquement, par contre les composants systèmes $CS_{1,1}$ et $CS_{2,1}$ ne le sont pas.

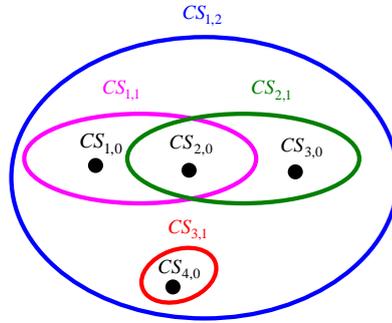


FIGURE 3.12 – Dispersion géographique

Propriété 4 - comportements émergent et coopératif : les composants systèmes $CS_{i',j'}$ d'un système $CS_{i,j}$ coopèrent si et seulement si pour tout $CS_{i',j'} \in CS_{i,j}$:

$$\bigcup_{\mathcal{M}_{i',j'}} \mapsto \mathcal{M}_{i,j}.$$

Les composants systèmes coopèrent afin de réaliser une mission globale $\mathcal{M}_{i,j}$ du SdS à partir d'un ensemble de missions $\mathcal{M}_{i',j'}$, que chaque composant système ne peut réaliser s'il est seul.

Le composant système $CS_{1,2}$ de la FIGURE 3.13 ne peut pas remplir ses missions M_4 et M_5 que par la coopération de ses composants systèmes $CS_{1,1}$ et $CS_{2,1}$ qui réalisent les missions M_1, M_2 et M_3 .

Propriété 5 - développement évolutionnaire : On peut élargir ou modifier la mission globale de notre SdS en introduisant ou, en éliminant des composants systèmes. Cette évolution correspond à l'ajout ou à la suppression de sommets et d'hyperarêtes de l'hypergraphe.

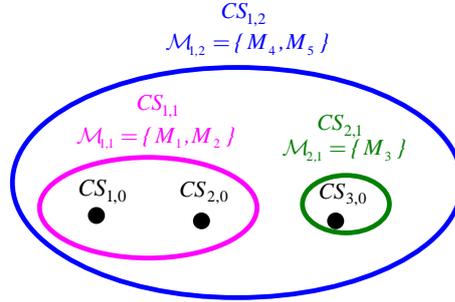


FIGURE 3.13 – Comportement émergent et coopératif

La FIGURE 3.14 montre l'hypergraphe \mathcal{HG}_2 déduit de l'hypergraphe \mathcal{HG}_1 après avoir supprimé les composants systèmes $CS_{4,0}$ et $CS_{6,0}$, tandis que \mathcal{HG}_3 représente l'ajout des composants systèmes $CS_{8,0}$ et $CS_{2,2}$ à \mathcal{HG}_2 .

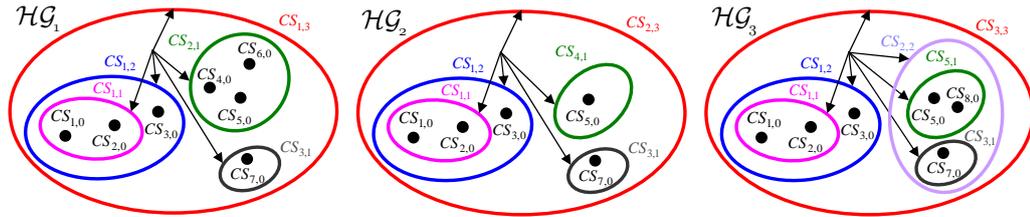


FIGURE 3.14 – Développement évolutif

En respectant les propriétés d'un SdS suivant la théorie des hypergraphes, nous constatons que le SdS se caractérise graphiquement par un hypergraphe imbriqué et valué comme le montre la FIGURE 3.15.

La FIGURE 3.15 donne l'exemple du système $CS_{1,3}$ constitué de sept composants systèmes de bas niveau :

$$CS_{1,0}, CS_{2,0}, CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0}, CS_{7,0},$$

et six composants systèmes indépendants et interconnectés :

$$CS_{1,1}, CS_{2,1}, CS_{3,1}, CS_{4,1}, CS_{1,2}, CS_{2,2},$$

représentés par les deux formalismes suivants :

– l'hyperensemble (à gauche)

$$\mathcal{HE} = \{ \{ \{ CS_{1,0}, CS_{2,0} \}, \{ CS_{3,0} \} \}, \{ \{ CS_{4,0}, CS_{5,0} \}, CS_{6,0} \}, \{ CS_{7,0} \} \},$$

- l'hypergraphe imbriqué et valué \mathcal{HG} (à droite), où :
 - $\mathcal{V} = \{CS_{1,0}, CS_{2,0}, CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0}, CS_{7,0}\}$
 - $\mathcal{E} = \{CS_{1,1}, CS_{2,1}, CS_{3,1}, CS_{4,1}, CS_{1,2}, CS_{2,2}, CS_{1,3}\}$
 - $\mathcal{M} = \{\mathcal{M}_{1,1}, \mathcal{M}_{2,1}, \mathcal{M}_{3,1}, \mathcal{M}_{4,1}, \mathcal{M}_{1,2}, \mathcal{M}_{2,2}, \mathcal{M}_{1,3}\}$

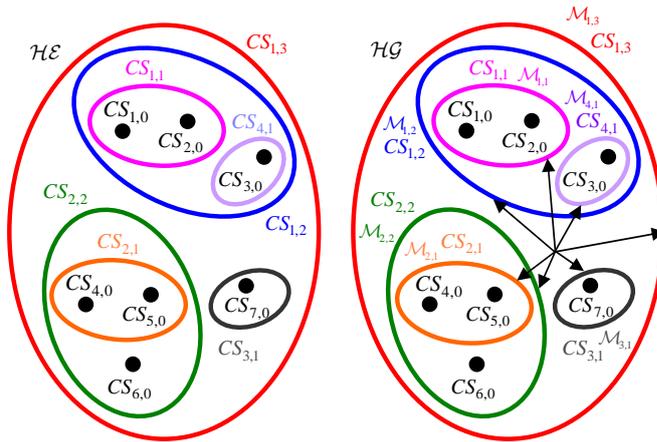


FIGURE 3.15 – Représentation du système $CS_{1,3}$ par un hyperensemble (à gauche) et par un hypergraphe imbriqué et valué (à droite)

La structure hiérarchique de l'hypergraphe imbriqué et valué $\mathcal{HG}(\mathcal{V}, \mathcal{E}, \mathcal{M})$ permet de représenter un SdS sur une échelle verticale multi-niveaux et une échelle transversale multi-structures comme le montre la FIGURE 3.16.

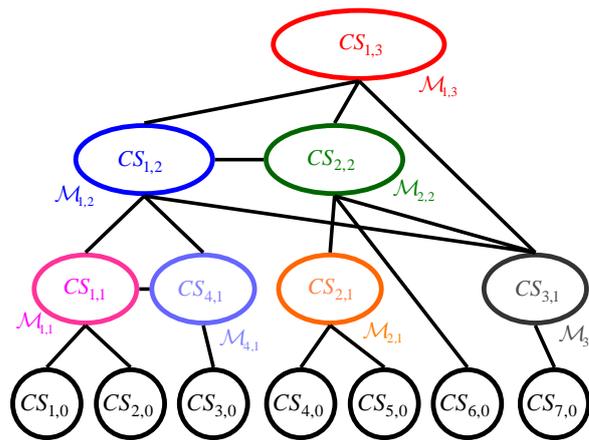


FIGURE 3.16 – représentation hiérarchique de l'hypergraphe imbriqué et valué de la FIGURE 3.15

La modélisation d'un SdS sous la forme d'hypergraphe permet de définir une architecture de supervision basée sur l'état de chaque composant système. Ainsi, en parcourant l'hypergraphe du SdS du bas vers le haut (FIGURE 3.16), nous pouvons voir comment un changement de mode de fonctionnement d'un composant système de niveau inférieur se répercute sur les ensembles de niveaux supérieurs. En parcourant l'hypergraphe du haut vers le bas, il est possible de trouver des solutions de reconfiguration optimales en réponse au changement d'état d'un composant système, tout en préservant les propriétés d'un SdS.

3.5 Supervision d'un système de systèmes à base d'un modèle hypergraphe

3.5.1 Architecture du système de supervision

Nous avons montré dans la section précédente qu'un SdS peut être représenté graphiquement par un hypergraphe imbriqué et valué. Nous avons démontré que ce type de modélisation permet de satisfaire les propriétés fondamentales d'un SdS, tout en décrivant une architecture hiérarchisée, multi-niveaux et multi-structures.

En exploitant les propriétés du modèle graphique du SdS, nous pouvons élaborer un schéma pour superviser un tel système. Ainsi, en surveillant le fonctionnement de chaque composant système, nous pouvons connaître le mode de fonctionnement global du SdS en fonction de son évolution et prévoir des solutions de reconfiguration lorsque des défaillances sont observées.

Définition 3.9 : La supervision d'un système est l'ensemble des outils et méthodes pour la gestion des modes de fonctionnement normal et défaillant ainsi que les moyens de leur reconfiguration. Les activités concernées par la supervision sont donc la détection et la localisation de défauts (surveillance) et la reconfiguration. Ces deux niveaux sont plus connus par la communauté automatique par la détection et localisation des défauts (Fault Detection and Isolation : FDI) et la commande tolérante aux fautes (Fault Tolerant Control : FTC).

Dans cette section, nous allons expliquer comment à partir d'un modèle graphique d'un SdS sous forme d'hypergraphe, nous obtenons un schéma de supervision en ligne, réalisé selon les étapes suivantes :

1. Surveillance des composants systèmes pour détecter et isoler des défaillances apparaissant durant leur fonctionnement. Cette étape est supposée déjà réalisée dans notre développement et représente une entrée du système de supervision.
2. Recherche du mode de fonctionnement du SdS en calculant le degré de satisfaction de contraintes associé. Pour cela, il est important d'affiner le niveau de fonctionnement en calculant les degrés de satisfaction de contraintes associés aux composants systèmes des niveaux inférieurs. En explorant de façon ascendante l'hypergraphe du SdS, nous pouvons suivre l'influence de la propagation par structure et par niveau, de la défaillance d'un composant système sur l'ensemble de SdS.
3. En présence de défaillances, reconfiguration du SdS suivant son mode de fonctionnement. En parcourant l'hypergraphe du SdS dans le sens descendant, il est possible de le reconfigurer tout en garantissant ses propriétés fondamentales. Cette reconfiguration permet de continuer la réalisation de la mission globale du SdS avec éventuellement des performances réduites.

3.5.2 Surveillance

Dans cette partie, nous nous intéressons principalement à la nature ainsi qu'au type de la variable dynamique, utilisée comme une entrée combinatoire et indicative du mode de fonctionnement d'un composant système $CS_{i,j}$. Si nous prenons le cas des systèmes physiques comme des composants systèmes élémentaires, par exemple : actionneurs électromécaniques, pompes, roues, suspension ... les entrées que nous cherchons pour notre modèle graphique de SdS sont les états défaillant (1) ou normal (0) de chaque composant système. Ces états, qui représentent la disponibilité de la fonction de l'équipement, sont fournis par des indicateurs de fautes. Ces derniers sont calculés grâce aux algorithmes de détection et d'isolation des défauts (FDI), qui consistent principalement à comparer le comportement réel du composant système avec son comportement en fonctionnement normal (détection de défaut), ou avec différents types de défauts (analyse et isolation des pannes), tout en minimisant les fausses alarmes, les non détections, et les retards dans la détection des défauts.

Plusieurs travaux ont été réalisés dans la littérature pour développer des méthodes de surveillance, en exploitant les informations disponibles pour décrire le comportement des composants systèmes. Deux types de méthodes sont ainsi proposées, les méthodes qualitatives et les méthodes quantita-

tives [TOSCANO, 2005]. L'approche qualitative consiste à décomposer l'espace paramétrique du modèle d'un composant système en plusieurs classes correspondant à des modes de fonctionnement connus, puis à déterminer par apprentissage les relations mathématiques entre les effets (observations des experts, mesures de capteurs et données statistiques), et les causes (défauts) [ROTSHEIN, 2005]. Les méthodes quantitatives, appelées aussi méthodes à base de modèles, consistent à générer des Relations de Redondances Analytiques (RRAs), qui expriment la différence entre les informations issues du composant système réel et celles générées par son modèle de fonctionnement normal. Ces relations appelées également résidus ou indicateurs de fautes, caractérisent suivant la précision du modèle, le mode de fonctionnement du système, et doivent être nulles lorsque le système ne présente aucune défaillance [BASSEVILLE, 1993], [ARMENGOL, 2000], [BLANKE, 2006].

Dans nos travaux, ce n'est pas la méthodologie du calcul qui nous intéresse, nous supposons que cette étape est correctement réalisée. Par contre, nous exploitons l'information issue des indicateurs afin d'élaborer une stratégie pour définir les modes de fonctionnement d'un SdS à partir de son modèle d'hypergraphe.

Toutefois, le présent mémoire étant réalisé dans le cadre du projet InTraDE, les outils utilisés pour le diagnostic sont basés sur les méthodes à base des modèles bond graph et fonctionnels développées dans les travaux de recherche de Nizar CHATTI qui représente une boîte à outils du système de supervision global du projet [CHATTI, 2011-a], [CHATTI, 2011-b].

3.5.3 Gestion des modes de fonctionnement sur l'hypergraphe

Détermination de mode de fonctionnement

Pour identifier le mode de fonctionnement d'un composant système $CS_{i,j}$, il est nécessaire d'associer à ce composant un problème de satisfaction de contraintes (CSP), noté $C_{i,j}$, permettant de définir le mode pour lequel le composant système opère [LYNCE, 2002], [GROHE, 2006]. Ce dernier dépend des entrées $e_{i,j}^k$ ainsi que des contraintes sur les composants systèmes $CS_{i',j'}$ de niveaux inférieurs de la manière suivante :

$$C_{i,j} = f(e_{i,j}^k(t), C_{i',j'}),$$

où $C_{i',j'}$ sont les CSP associés aux composants systèmes $CS_{i',j'}$ tels que $CS_{i',j'} \in CS_{i,j}$.

Nous associons à chaque $C_{i,j}$ un degré de satisfaction $\tau_{i,j}$ défini comme suit :

$$\tau_{i,j} : \mathcal{C} \times \mathbb{R}^+ \mapsto \mathbb{T}$$

$$(C_{i,j}, t) \mapsto \tau_{i,j} \begin{cases} = \tau_{min} & \text{si pas satisfait,} \\ = \tau_{max} & \text{si totalement satisfait,} \\ \in]\tau_{min}, \tau_{max}[& \text{si partiellement satisfait.} \end{cases}$$

Le degré de satisfaction $\tau_{i,j}$ permet de situer le niveau de fonctionnement du composant système $CS_{i,j}$ lorsque les contraintes de $C_{i,j}$ sont partiellement satisfaites. Pour calculer $\tau_{i,j}$, il est indispensable de connaître le degré de satisfaction $\tau_{i',j'}$ de chaque composant système $CS_{i',j'} \in CS_{i,j}$ ainsi qu'un ensemble d'indicateurs dynamiques $e_{i,j}^k(t)$. Une fois que le degré de satisfaction $\tau_{i,j}$ du composant système $CS_{i,j}$ est calculé, le mode de fonctionnement associé est alors déduit.

Soit $Mod_{i,j}$ l'ensemble fini des modes de fonctionnement du système $CS_{i,j}$, et \mathcal{S} une fonction qui associe à chaque instance $(e_{i,j}^k(t), C_{i',j'})$ un mode unique $mod \in Mod_{i,j}$. La fonction \mathcal{S} , n'est pas injective, c'est-à-dire que, le même mode peut être l'image de plusieurs instances.

Le mode de fonctionnement du système $CS_{i,j}$ est dit « normal », si les indicateurs $e_{i,j}^k(t)$ ainsi que toutes les contraintes de $C_{i',j'}$ dont il dépend sont totalement satisfaits. Il est dit « dégradé » si certains indicateurs et/ou contraintes sont partiellement satisfaits, et il est dit « arrêt » si les indicateurs et les contraintes ne sont pas satisfaits.

Prenons par exemple le composant système $CS_{1,2}$ composé de $CS_{1,1}$ et $CS_{4,1}$ (FIGURE 3.18). $CS_{1,2}$ dépend des deux CSP $C_{1,1}$ et $C_{4,1}$ et des entrées $e_{1,2}^k$. Le mode « 111 » est un mode « normal », et « 000 » est un mode « arrêt » alors que la zone grise représente les modes « dégradés » (FIGURE 3.17).

La transition entre les différents modes de fonctionnement est définie par un automate fini déterministe comme suit :

$$\Sigma = \langle Mod_{i,j}, \Delta_{i,j}, \delta \rangle,$$

où $Mod_{i,j}$ est l'ensemble fini des modes de fonctionnement du système $CS_{i,j}$, $\Delta_{i,j}$ est l'ensemble fini de toutes les instances possibles $(e_{i,j}^k(t), C_{i',j'})$, et δ est la fonction de transition définie comme suit :

$$\delta : Mod_{i,j} \times \Delta_{i,j} \mapsto Mod_{i,j}.$$

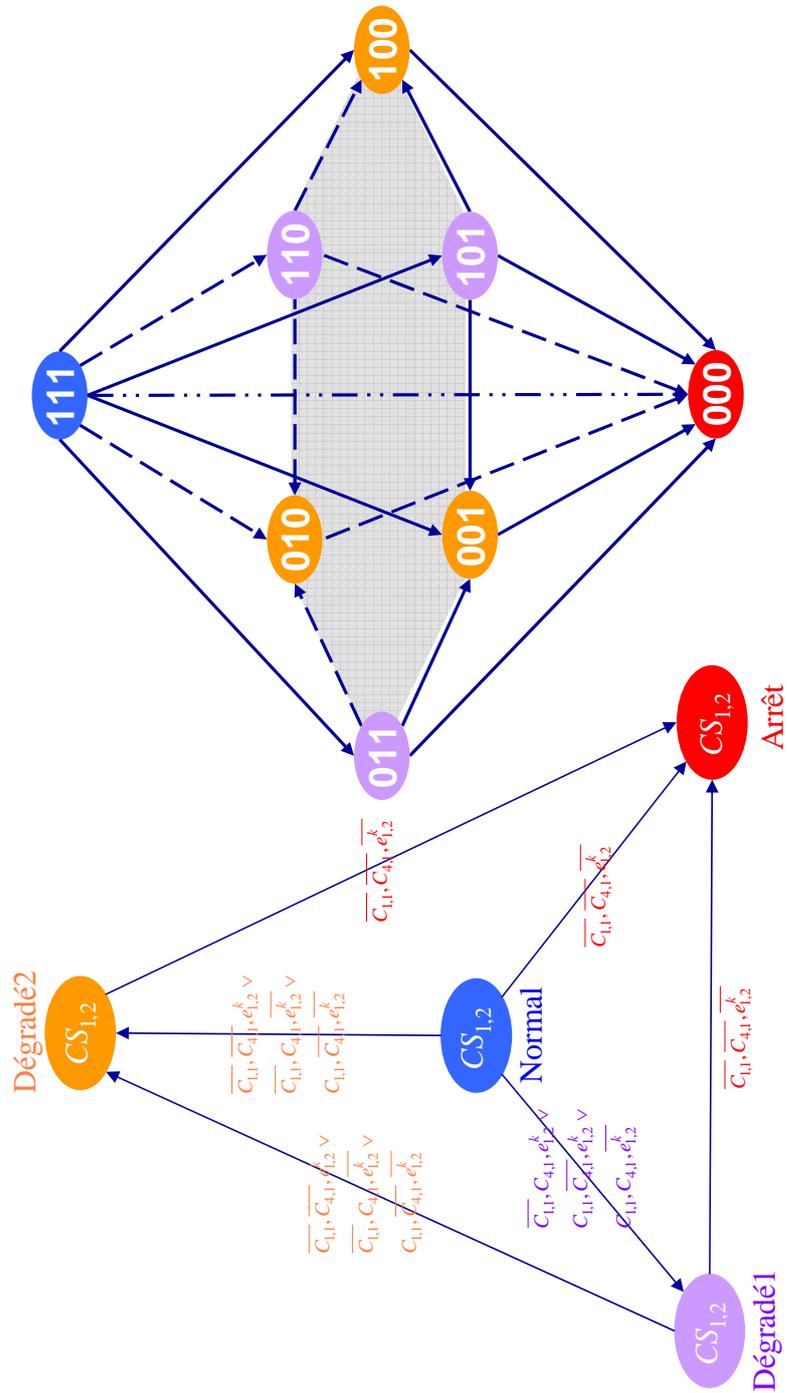


FIGURE 3.17 – Modes de fonctionnement du système $CS_{1,2}$

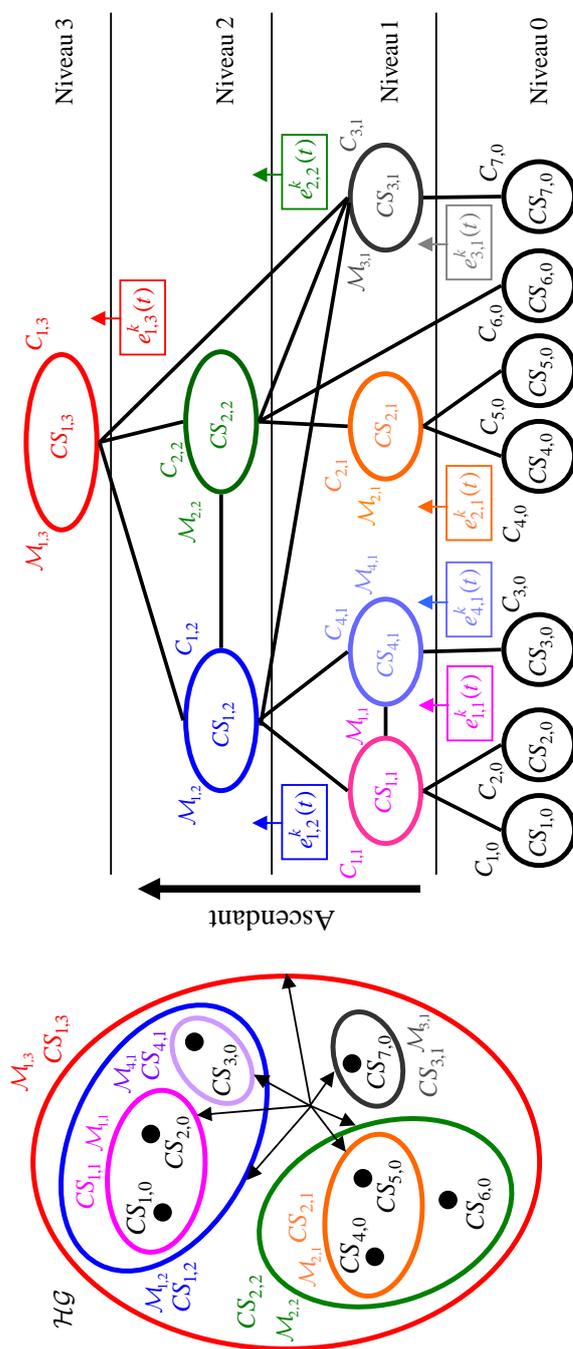


FIGURE 3.18 – Représentation hiérarchique d'un hypergraphe imbriqué et valué

Parcours ascendant de l'hypergraphe pour la surveillance

La représentation hiérarchique d'un SdS sous forme d'un hypergraphe imbriqué et valué permet de suivre les conséquences de la défaillance d'un composant système sur l'ensemble du SdS en parcourant le graphe de façon ascendante (FIGURE 3.18).

Dans l'exemple de la FIGURE 3.18, nous pouvons constater que l'hypergraphe associé est décrit par un triplet $(\mathcal{V}, \mathcal{E}, \mathcal{M})$ comme suit :

$$\begin{aligned}\mathcal{V} &= \{CS_{1,0}, CS_{2,0}, CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0}, CS_{7,0}\}, \\ \mathcal{E} &= \{CS_{1,1}, CS_{2,1}, CS_{3,1}, CS_{4,1}, CS_{1,2}, CS_{2,2}, CS_{1,3}\}, \\ \mathcal{M} &= \{\mathcal{M}_{1,1}, \mathcal{M}_{2,1}, \mathcal{M}_{3,1}, \mathcal{M}_{4,1}, \mathcal{M}_{1,2}, \mathcal{M}_{2,2}, \mathcal{M}_{1,3}\}.\end{aligned}$$

Lorsqu'un composant système tel que $CS_{4,1}$ est défini comme composant défaillant (FIGURE 3.19) à la suite de l'étape 1, nous constatons que les composants systèmes $CS_{1,1}$ et $CS_{1,2}$ du même niveau et d'un niveau différent sont affectés, directement pour $CS_{1,2}$ et indirectement pour $CS_{1,1}$. Ainsi la mission globale $\mathcal{M}_{1,2}$ du composant système $CS_{1,2}$ peut être interrompue ou réalisée d'une façon moins performante qu'à l'initial suivant le degré de défaillance.

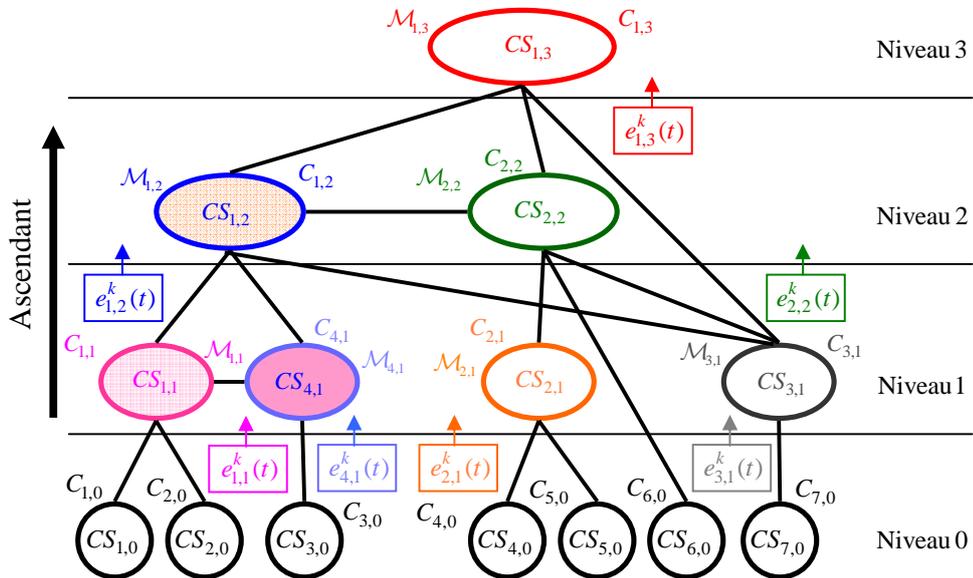


FIGURE 3.19 – Influence de $CS_{4,1}$ sur $CS_{1,2}$ et $CS_{1,1}$

La propagation de l'effet de la défaillance suit son parcours ascendant en af-

fectant le composant système $CS_{1,3}$ dont la mission $\mathcal{M}_{1,3}$ représente la tâche globale du SdS. La FIGURE 3.20 montre la propagation de l'effet de la défaillance du système $CS_{4,1}$ sur l'ensemble du SdS.

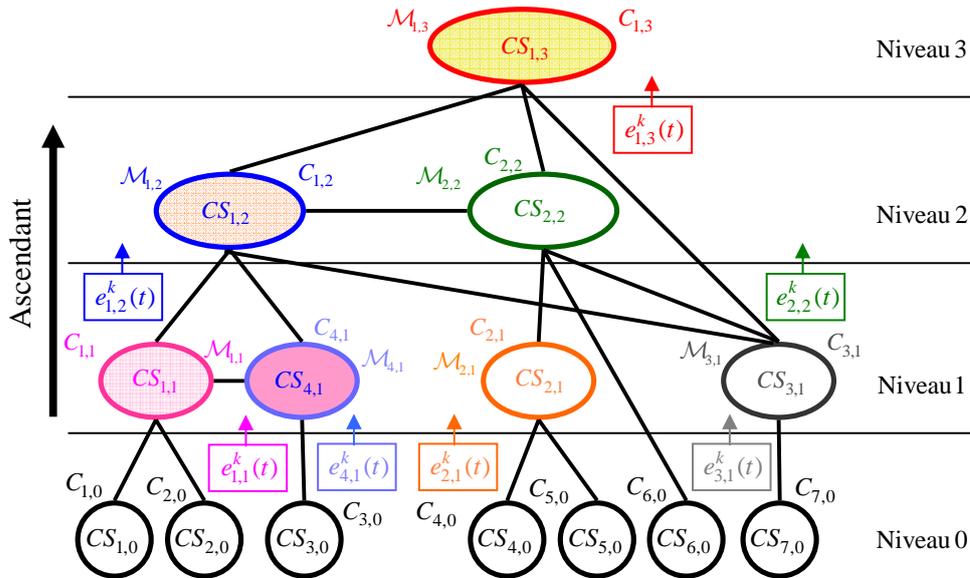


FIGURE 3.20 – Propagation de l'effet de la défaillance sur le SdS

Ainsi, la sensibilité des composants systèmes du SdS à une défaillance d'un composant de niveau 1 peut être représentée par la matrice de défaillance de la TABLE 3.1. La présence de « 1 » sur une colonne associée à un composant système $CS_{i,1}$ se traduit par l'influence de la défaillance de ce composant système sur les autres composants systèmes de même niveau ou de niveaux différents. Tandis que la présence de « 0 » montre l'insensibilité à la défaillance du composant système.

3.5.4 Parcours descendant de l'hypergraphe pour la reconfiguration

En parcourant le graphe, selon le sens descendant plutôt que de manière ascendante (sens utilisé pour déterminer le mode de fonctionnement), il devient possible d'envisager des possibilités de reconfiguration du SdS en fonction de la défaillance détectée. Cette reconfiguration est réalisée en résolvant le CSP correspondant au SdS. Résoudre un CSP consiste à résoudre, dans ce cas, un problème d'optimisation vertical (multi-niveaux), qui permet de trouver une

		$CS_{1,1}$	$CS_{2,1}$	$CS_{3,1}$	$CS_{4,1}$
$CS_{1,1}$	$(C_{1,1})$	1	0	0	1
$CS_{2,1}$	$(C_{2,1})$	0	1	0	0
$CS_{3,1}$	$(C_{3,1})$	0	0	1	0
$CS_{4,1}$	$(C_{4,1})$	1	0	0	1
$CS_{1,2}$	$(C_{1,2})$	1	0	1	1
$CS_{2,2}$	$(C_{2,2})$	0	1	1	0
$CS_{1,3}$	$(C_{1,3})$	1	1	1	1

TABLE 3.1 – Matrice de défaillance vue du niveau 1

solution minimisant le poids des contraintes violées (non satisfaites). Cela se fait, en affectant à chacune des variables du CSP une valeur de son domaine de façon à ce que, aucune combinaison de valeurs non autorisées par les contraintes n'apparaisse. Cette affectation peut être soit totale, c'est-à-dire qu'elle ne viole aucune contrainte, soit partielle dans le cas contraire.

Un bon nombre d'algorithmes permettant de résoudre des CSP sont développés dans la littérature [ALLIOT, 2002], [DECHTER, 2003], [VERFAILLIE, 2005], [MERTENS, 2006], [DECHTER, 2007]. L'algorithme le plus courant permettant d'explorer un arbre de recherche est celui du backtracking (retour en arrière chronologique) [ALLIOT, 2002], [DECHTER, 2003]. Celui-ci résout un CSP en trouvant petit à petit une solution partielle, qui tend vers une solution complète, en commençant par réaliser un parcours en profondeur de l'arbre de recherche. Si l'instanciation d'une variable mène à la non satisfaction d'une contrainte, alors il est inutile de continuer à explorer le sous-arbre enraciné dans le nœud qui succède immédiatement à cette instanciation. L'algorithme de backtracking revient alors sur sa décision précédente et continue l'exploration en profondeur (en reprenant) à partir d'une nouvelle instanciation.

Afin de rendre le backtracking encore plus performant, différentes approches ont été mises en œuvre pour réduire le domaine de recherche des variables non encore instanciées [ALLIOT, 2002], [DECHTER, 2003], [STEFANOVITCH, 2010]. En effet, il est possible, avant d'exécuter un algorithme de résolution, de déterminer que pour un sous-problème particulier, certaines valeurs des variables ne peuvent jamais mener à une solution. Il est alors intéressant d'éliminer ces valeurs du domaine des variables concernées afin de réduire le facteur de branchement de l'arbre de recherche et ainsi diminuer la complexité de la recherche.

De nos jours, plusieurs solveurs de contraintes performants, dits aussi langages de modélisation, sont développés (Gecode, Choco, Sat4j, AMPL ...). Le fonctionnement général d'un solveur est de donner dans le meilleur des cas une solution résolvant le CSP $C_{i,j}$ à partir d'un ensemble d'entrées $C_{i',j'}$ et $e_{i,j}^k(t)$ qui ont chacune leur propre domaine. Si le problème ne peut pas être satisfait, le solveur indique que la reconfiguration n'est pas possible.

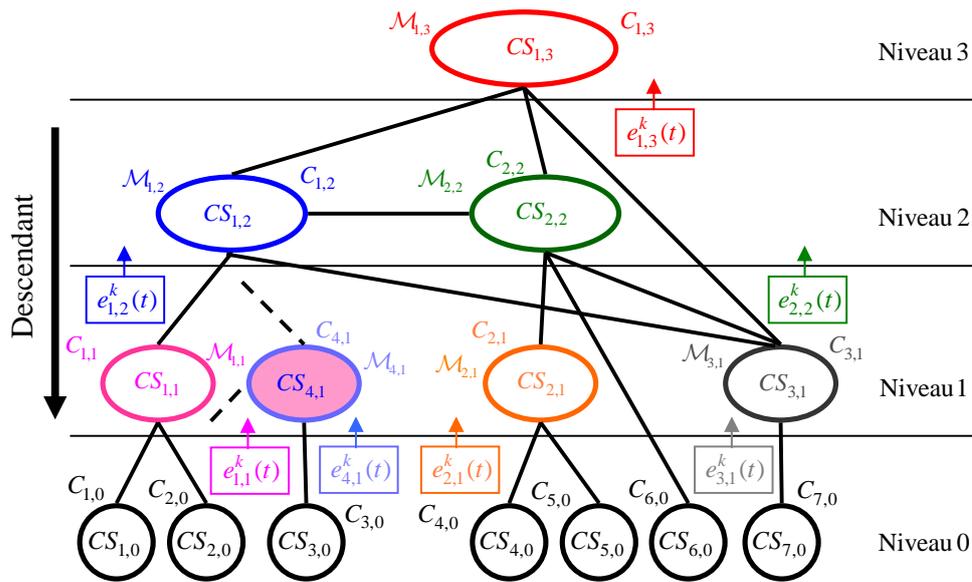


FIGURE 3.21 – Reconfiguration du système $CS_{1,3}$ sans le composant système $CS_{4,1}$

Dans notre travail, ce n'est ni le développement de nouveaux algorithmes, ni l'étude de la complexité de ceux qui existent qui nous intéresse. Nous supposons que cette étape est bien établie. Par contre, nous exploitons la solution issue d'un solveur afin d'élaborer une stratégie de reconfiguration, s'appuyant sur la liste des composants systèmes dont le mode de fonctionnement est dégradé. Pour ce faire, les composants systèmes dont le mode de fonctionnement est dégradé et les liens associés sont supprimés de l'hypergraphe modélisant le SdS. Dans le cas où un ou plusieurs composants systèmes seraient utilisés pour palier le composant défaillant, ils seraient, ainsi que liens correspondants, ajoutés à l'hypergraphe. À la suite de l'étape 1, le composant système $CS_{4,1}$ est détecté défaillant. L'étape 2 détermine que son mode de fonctionnement est « arrêt ». Le composant système est dissocié du SdS en le supprimant avec ses liens (FIGURE 3.21). Le nouveau SdS doit préserver les mêmes propriétés graphiques que son précédent.

3.6 Application à un système de transport intelligent

Le système de transport intelligent (ITS) est défini comme un ensemble de composants systèmes (système de transport), opérant dans un environnement dynamique et pouvant coopérer autour de tâches spécifiques telles que la gestion de trafic, l'ordonnancement ...

Les échanges d'informations sont possibles entre les différents systèmes de transport par l'utilisation d'outils des systèmes ICT, par exemple, le système de géo-localisation, les capteurs intelligents de distance et de détection d'obstacles ...

Le transport intelligent apparait souvent dans des environnements privés, où l'espace d'exploitation reste confiné et sécurisé. Dans ce type d'espace, l'environnement est souvent connu, mieux que pour un espace urbain, permettant ainsi d'améliorer les performances de contrôle et de supervision du système ITS.

Depuis la fin de la Seconde Guerre mondiale, les concepts de transport à l'intérieur des zones confinées se modernisent et se diversifient. Les mêmes problématiques se trouvent quelque soit l'application, port [REVIEW], aéroport [MUKHERJEE, 2004], espace urbain [DARPA]. Améliorer la gestion du trafic dans des espaces confinés, en conservant les infrastructures existantes, peut être possible grâce à l'utilisation d'un système ITS. Ce dernier exige la mise en place de nouvelles solutions technologiques visant à améliorer la fluidité du trafic tout en respectant les règles environnementales et sécuritaires [LE-ANH, 2006].

Un tel système de transport différencie clairement les indépendances opérationnelle et managériale de chacun de ses composants lorsque le niveau élémentaire correspond aux systèmes de transport. Il permet de faire coopérer ses composants systèmes transport, dispersés géographiquement tout en restant évolutionnaire suivant les missions demandées.

Prenons l'exemple d'un système intelligent situé dans un terminal portuaire de la FIGURE 3.22. Ce dernier est composé de trois stations de chargement et déchargement, et d'un réseau routier.

Dans cet environnement dynamique et confiné, nous considérons des compo-



FIGURE 3.22 – Vue aérienne du terminal maritime Radicatel

sants systèmes pour le transport du fret. Ils correspondent à un train⁴ de véhicules intelligents autonomes (IAVs) de la FIGURE 3.23 conçus dans le cadre du projet [InTraDE].



FIGURE 3.23 – Train de trois véhicules intelligents autonomes

4. train de véhicules est appelé aussi « platoon », où les véhicules ne sont pas connectés physiquement et se suivent en formant un train de véhicules autonomes.

Chaque IAV (FIGURE 3.24) est configuré en mode de conduite 4×4 avec actionnement décentralisé [MERZOUKI, 2007]. Il peut être piloté manuellement ou automatiquement, et roule tout en s'adaptant aux changements de l'environnement qui l'entoure. La perception de l'environnement est assurée par différents systèmes de mesures embarqués qui opèrent en temps réel.

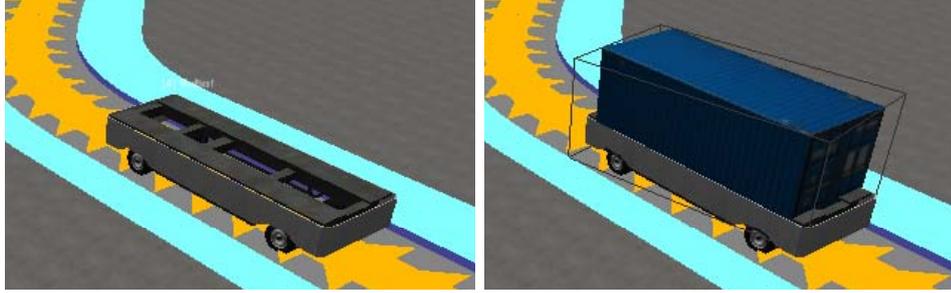


FIGURE 3.24 – Composant système IAV

À travers cet exemple, où le système ITS peut être considéré comme un SdS, nous reprenons les points méthodologiques vus précédemment pour montrer comment l'hypergraphe est un réel outil d'aide à la supervision des SdS.

3.6.1 Modélisation du système de transport intelligent

Le système ITS composé de stations, de véhicules autonomes, et d'une infrastructure routière peut être représenté par un hypergraphe imbriqué et valué $\mathcal{HG}(\mathcal{V}, \mathcal{E}, \mathcal{M})$ où :

$$\mathcal{V} = \{Ac_1^1, Ac_1^2, Ac_1^3, Ac_1^4, Ac_2^1, Ac_2^2, Ac_2^3, Ac_2^4, Ac_3^1, Ac_3^2, Ac_3^3, Ac_3^4\},$$

représente l'ensemble des actionneurs de traction de trois véhicules. Nous supposons dans cet exemple, que le niveau élémentaire du système ITS est composé uniquement par des actionneurs.

$$\mathcal{E} = \{IAV_1, IAV_2, IAV_3, St_1, St_2, St_3, Train, Stations, Réseau routier, ITS\},$$

représente l'ensemble des hyperarêtes correspondent aux composants systèmes quelque soit leur niveau, supérieur à élémentaire (niveau 0), ainsi que leurs liens.

\mathcal{M} est l'ensemble des missions associées aux composants systèmes et au système ITS.

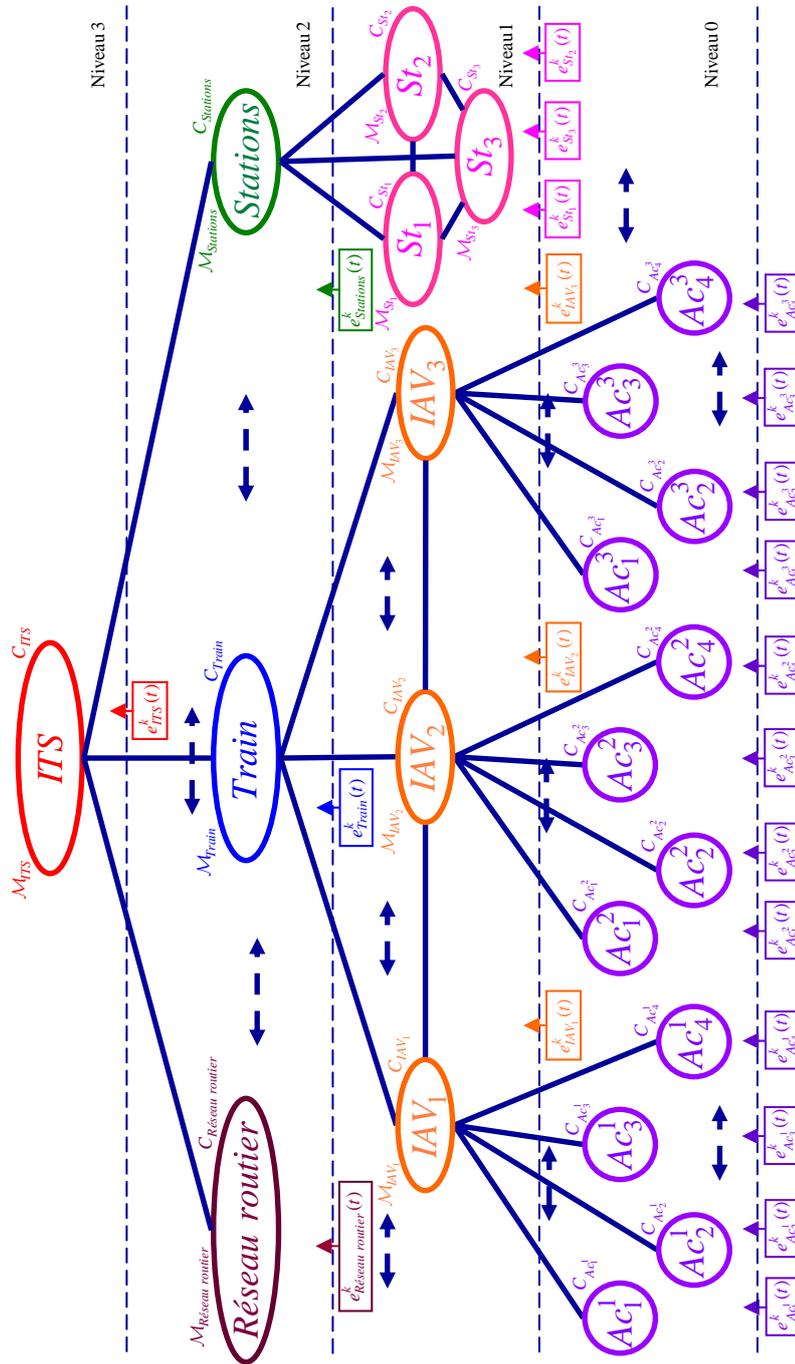


FIGURE 3.25 – Hypergraphe imbriqué et valué d'un ITS : représentation hiérarchique

Une représentation ensembliste et hiérarchique du modèle hypergraphe du système ITS est donnée par les deux FIGURES 3.25 et 3.26.

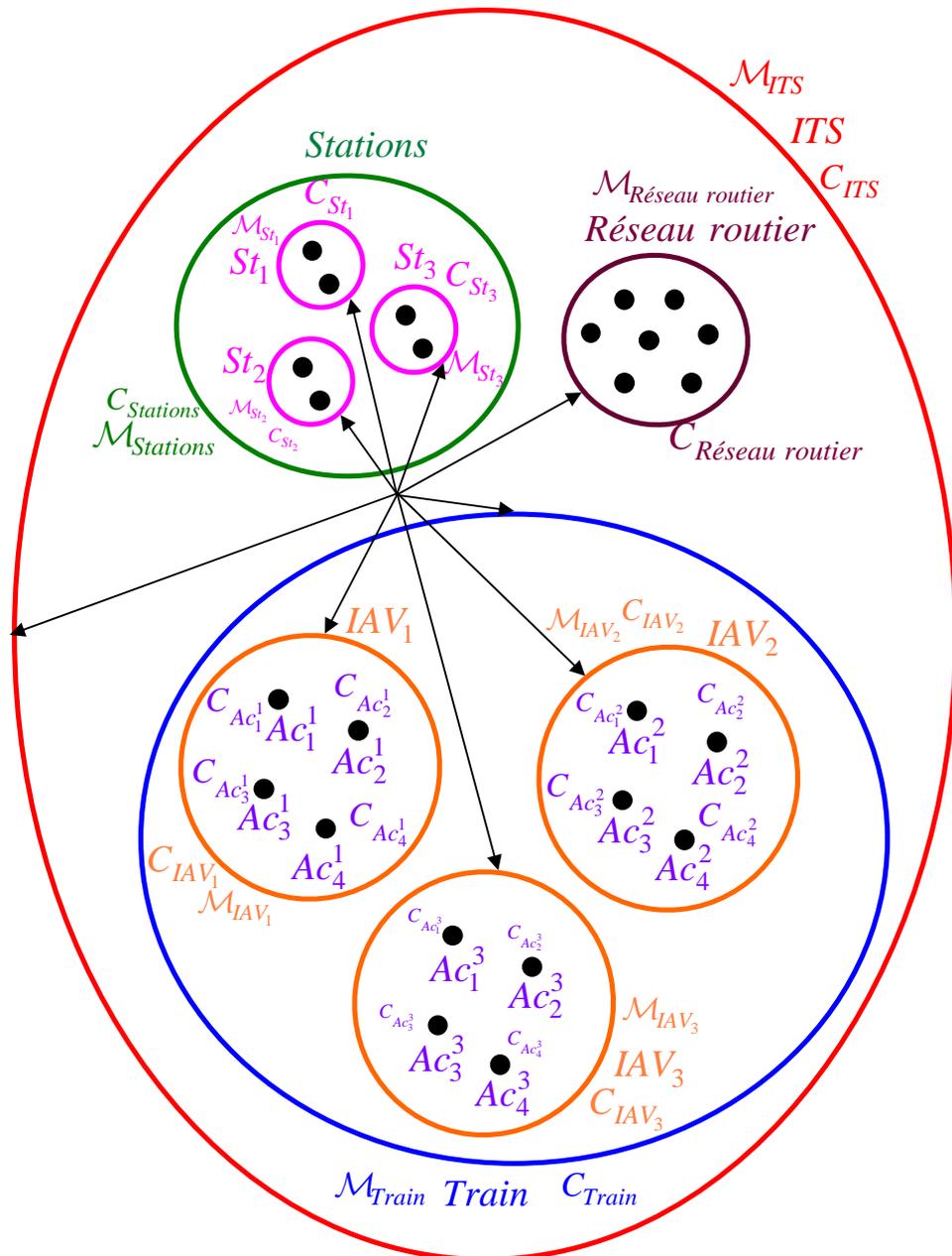


FIGURE 3.26 – Hypergraphe imbriqué et valué d'un ITS : représentation ensembliste

On peut vérifier que les propriétés du système ITS sont respectées. En effet on a :

1.

$$\forall CS_{i,j} \in ITS : CS_{i,j} \in \mathcal{E},$$

ainsi chaque composant système de ce système ITS est indépendant opérationnellement.

2.

$$\forall CS_{i',j'}, CS_{i'',j''} \in ITS : \mathcal{M}_{i',j'} \cap \mathcal{M}_{i'',j''} = \emptyset,$$

ainsi chaque composant système de ce système ITS est indépendant managérialement.

3.

$$\forall CS_{i',j'}, CS_{i'',j''} \in ITS : CS_{i',j'} \cap CS_{i'',j''} = \emptyset,$$

ainsi, il n'existe pas de couplage physique entre les composants systèmes. Ces derniers sont donc bien géographiquement dispersés.

4.

$$\forall CS_{i',j'} \in ITS : \bigcup_{\mathcal{M}_{i',j'}} \mapsto \mathcal{M}_{ITS},$$

ainsi la mission globale du système ITS est réalisée une fois que les missions de l'ensemble des composants systèmes sont terminées.

5. A chaque instant, le système ITS peut changer de structure selon le changement de mode de fonctionnement des composants systèmes ou si la mission change.

3.6.2 Surveillance

Dans cet exemple, nous surveillons le mode de fonctionnement des composants systèmes élémentaires de niveau 0 (actionneurs) grâce à des indicateurs de défauts calculés à partir des RRAs. Ces indicateurs de défaut expriment la différence entre les informations fournies par le système actuel et celles fournies par son modèle de fonctionnement normal [STAROSWIECKI, 2001].

Dans le cas du véhicule intelligent autonome, les indicateurs de fautes, appelés résidus $r_i^{j,1}(t)$, $r_i^{j,2}(t)$, et $r_i^{j,3}(t)$, du $j^{\text{ème}}$ actionneur ($j \in \{1, \dots, 4\}$) du $i^{\text{ème}}$ IAV ($i \in \{1, \dots, 3\}$) sont calculés comme suit [MERZOUKI, 2009] :

Partie électrique :

$$r_i^{j,1} = U_{0_i}^j - L_i^j \frac{dI_i^j}{dt} - R_{e_i}^j I_i^j - k_{e_i}^j \dot{\theta}_{e_i}^j$$

Partie mécanique :

$$r_i^{j,2} = k_{e_i}^j I_i^j - J_{e_i}^j \frac{d\theta_{e_i}^j}{dt} - f_{e_i}^j \dot{\theta}_{e_i}^j - w_i^j - K_i^j (\theta_{e_i}^j - N_{0_i}^j \theta_{s_i}^j)$$

Roue avec un contact au sol :

$$r_i^{j,3} = N_{0_i}^j w_i^j + N_{0_i}^j K_i^j (\theta_{e_i}^j - N_{0_i}^j \theta_{s_i}^j) - J_{s_i}^j \frac{d\theta_{s_i}^j}{dt} - f_{s_i}^j \dot{\theta}_{s_i}^j - R_i^j F_{x_i}^j$$

où $U_{0_i}^j$ est la tension d'alimentation, I_i^j est l'intensité mesurée du courant électrique, $N_{0_i}^j$ est la constante de réduction, $\theta_{e_i}^j$, $\theta_{s_i}^j$, $\dot{\theta}_{e_i}^j$, $\dot{\theta}_{s_i}^j$ sont respectivement les $j^{ème}$ positions mesurées et vitesses estimées de l'axe du moteur et de la roue, K_i^j est la raideur de la transmission, R_i^j est le rayon statique de la roue, $J_{e_i}^j$, $J_{s_i}^j$, $f_{e_i}^j$, $f_{s_i}^j$ sont les inerties et coefficients de frottement visqueux de l'actionneur et de la charge, w_i^j est le couple perturbateur, $R_{e_i}^j$, L_i^j , $k_{e_i}^j$ sont respectivement la résistance, l'inductance et la constante électrique, $F_{x_i}^j$ est l'effort longitudinal du contact roue-sol, du $j^{ème}$ actionneur du $i^{ème}$ véhicule.

En fonctionnement normal, les valeurs des résidus $r_i^{j,1}(t)$, $r_i^{j,2}(t)$, et $r_i^{j,3}(t)$ varient à l'intérieur de seuils calculés en fonction des incertitudes paramétriques. En présence de défauts, ces valeurs sortent de leurs seuils.

3.6.3 Identification des modes de fonctionnement

Soient $\mathcal{C} = \{C_{i,j}\}$ l'ensemble fini des contraintes du système ITS, et $Mod_{i,j}$ l'ensemble fini des modes de fonctionnement associés à chaque composant système $CS_{i,j}$. Nous constatons que pour le système global ITS, il existe cinquante deux contraintes (FIGURE 3.25) dont douze contraintes représentant les modes de fonctionnement des actionneurs (niveau 0).

L'état de chaque actionneur est défini par les valeurs issues de ses trois indicateurs de fautes (FIGURE 3.27), où les seuils λ_{1min} , λ_{1max} , λ_{2min} , λ_{2max} , λ_{3min} , et λ_{3max} sont calculés dans [DJEZIRI, 2009].

Les six contraintes du niveau 1 représentent les modes de fonctionnement actualisés des composants systèmes IAVs, et des stations de chargement ou déchargement. Nous considérons trois modes de fonctionnement pour chaque composant système IAV : « normal » IAV_{i_N} , « dégradé » IAV_{i_D} et « arrêt » IAV_{i_A} . Un composant système IAV est considéré en mode « dégradé », lorsque plusieurs actionneurs sont endommagés, alors qu'un composant système IAV endommagé est un véhicule complètement arrêté. Le composant système IAV

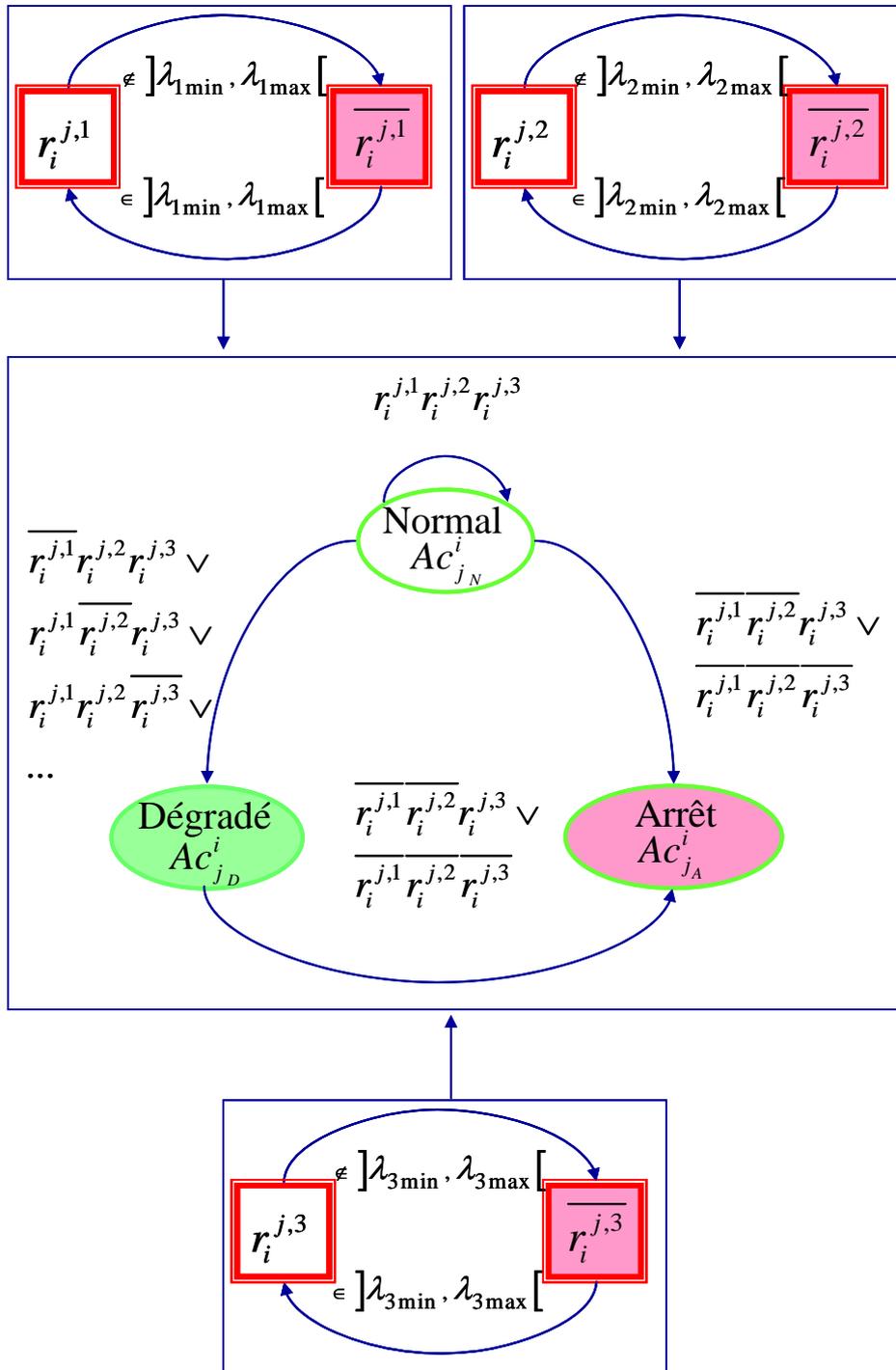


FIGURE 3.27 – Modes de fonctionnement des actionneurs

est capable de se reconfigurer en choisissant une combinaison d'actionneurs non endommagés. Le mode de fonctionnement d'une station peut être occupée, libre, ou saturée Les trois contraintes du niveau 2 représentent le mode de fonctionnement du train de trois IAVs, des stations, et du réseau routier. Un train de véhicules peut être en situation : « normal », « dégradé » ou « arrêt » (FIGURE. 3.28).

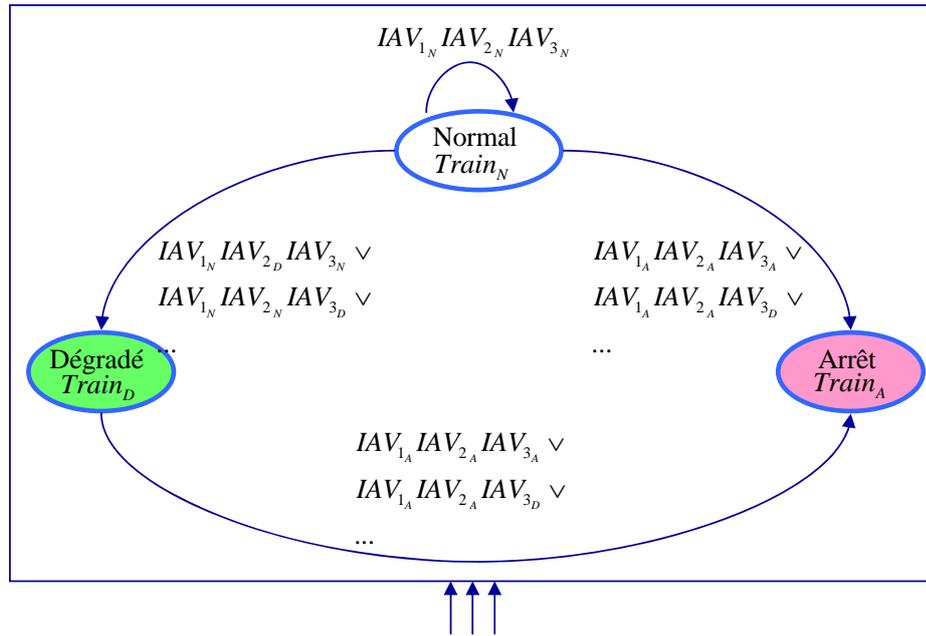


FIGURE 3.28 – Modes de fonctionnement du train de IAVs

Considérons maintenant que le train de véhicules IAVs est en mode de fonctionnement « normal » (FIGURE 3.25), où l'inter-distance maximale qui peut être détectée entre les IAVs est de 10 mètres. Pour éviter la collision, chaque IAV doit être distant du suivant d'au moins 2 mètres.

Le parcours ascendant de l'hypergraphe (FIGURE 3.29) permet de surveiller le fonctionnement des actionneurs Ac_j^i : les véhicules IAVs roulent en vitesse constante de 5 mètres/seconde en parcourant un chemin rectiligne. On introduit 13 secondes après le début de la simulation, des défauts sur la source d'alimentation et les courants électriques des 1^{er} et 3^{ème} actionneurs du 2^{ème} IAV. La surveillance de ces actionneurs permet de détecter ces défauts à 15 secondes, puis de les isoler à 18,5 secondes.

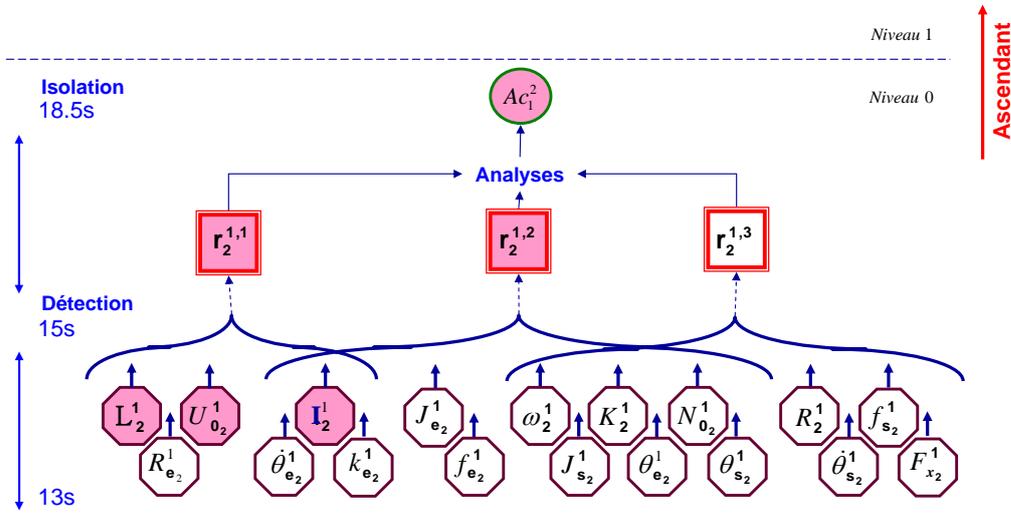


FIGURE 3.29 – Analyse du mode de fonctionnement de l’actionneur Ac_1^2 .

En suivant le parcours ascendant de l’hypergraphe (FIGURE 3.30), nous constatons que notre système ITS ne satisfait plus les contraintes associées aux composants systèmes.

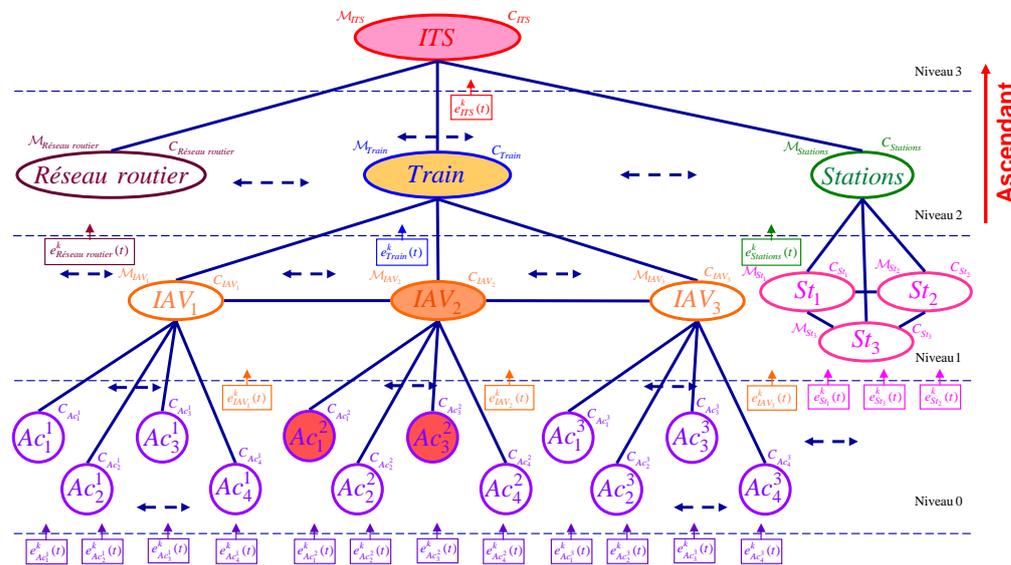


FIGURE 3.30 – ITS en mode « dégradé »

Le mode de fonctionnement du 2^{ème} IAV devient « arrêt », et l'automate suivant est appliqué :

$$\delta : Train_N \times IAV_{1N} IAV_{2A} IAV_{3N} \mapsto Train_D,$$

où $Train_N$ et $Train_D$ sont les états du train en mode « normal » et « dégradé ».

Pour éviter la collision avec le 2^{ème} IAV, le 3^{ème} véhicule suiveur s'arrête à l'instant 14 secondes.

3.6.4 Reconfiguration

La reconfiguration est réalisée en suivant le parcours descendant de l'hypergraphe. Le scénario envisagé, après la détection de l'arrêt du véhicule IAV_2 , consiste à sortir cet IAV du train de véhicule pour qu'il ne soit plus sur la trajectoire principale (FIGURE 3.31).

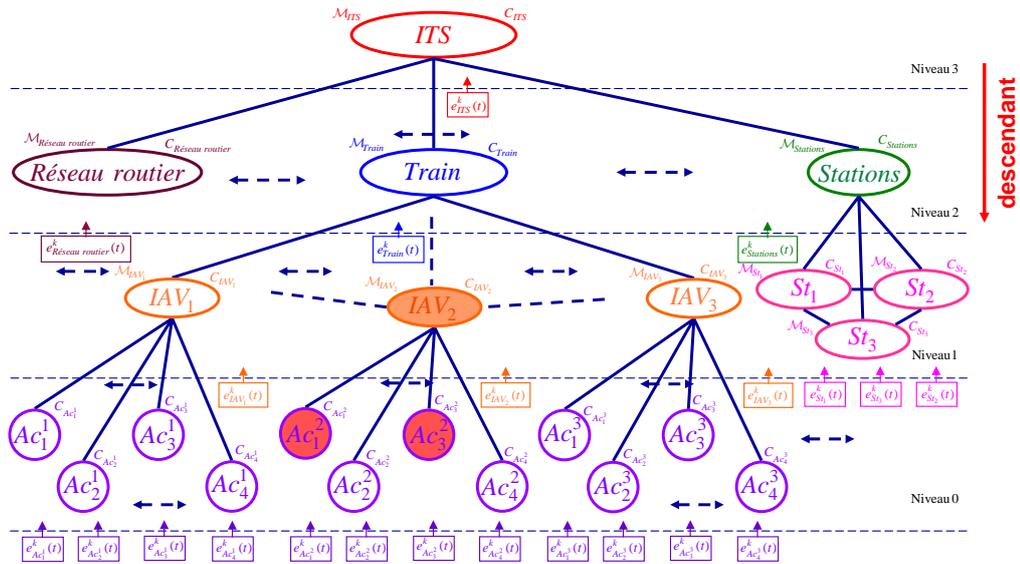


FIGURE 3.31 – Suppression de IAV_2 du train

La reconfiguration du système ITS (FIGURE 3.32) est effectuée par un enchaînement de reconfigurations du train de véhicules comme suit :

1. Reconfiguration 1 : arrêt du train et sortie du 2^{ème} IAV après une translation latérale, moyennant ses deux systèmes de direction décentralisés.

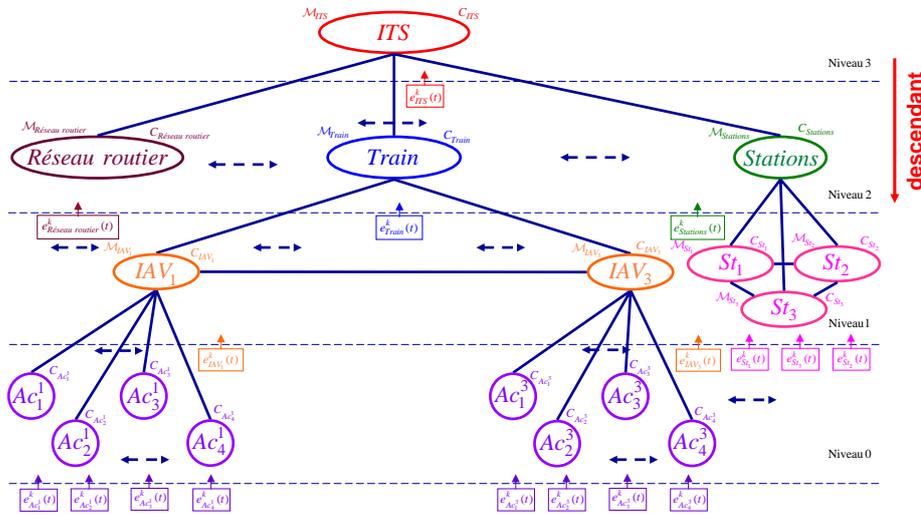


FIGURE 3.32 – Le train après la reconfiguration

2. Reconfiguration 2 : déplacement du 3^{ème} IAV pour se rapprocher du 1^{er} afin de reconstruire le nouveau train en respectant l'inter-distance à partir de l'information issue du capteur de distance frontal.
3. Reconfiguration 3 : reprise de la circulation du train de composants systèmes IAVs, reconfiguré pour qu'il puisse terminer la mission initiale avec moins de performances (deux IAVs au lieu de trois).

Le scénario étudié est décrit par les FIGURES 3.33 et 3.34.

3.7 Conclusion

Dans ce chapitre, nous avons présenté une modélisation graphique par l'approche de l'hypergraphe d'un Système de Systèmes (SdS). Le modèle graphique obtenu n'est qu'un hypergraphe imbriqué et valué, répondant aux propriétés fondamentales des SdS, à savoir l'indépendance, la coopération, l'évolution, et la dispersion.

L'absence de calcul sur un tel modèle graphique permet une représentation du multi-niveaux et multi-structures du SdS. Nous pouvons identifier le mode de fonctionnement des composants systèmes ainsi que celui du SdS en parcourant de manière ascendante le modèle de l'hypergraphe. Dans le sens descendant, nous pouvons identifier des reconfigurations possibles du SdS pour maintenir la réalisation de la mission globale. L'hypergraphe apparaît alors comme un outil permettant d'établir une architecture de supervision

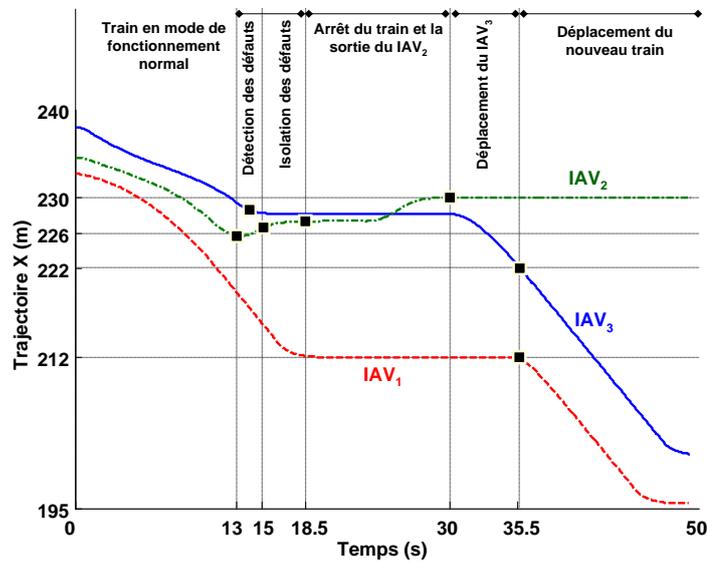


FIGURE 3.33 – Mode de fonctionnement de trois IAVs

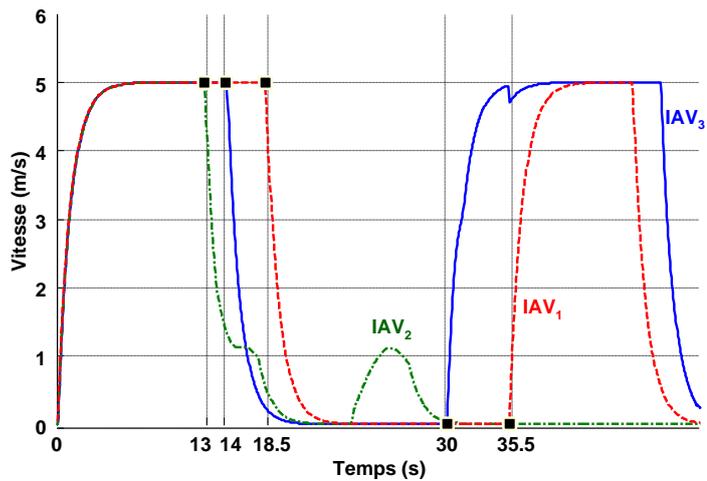


FIGURE 3.34 – Vitesse réelle de trois IAVs

d'un SdS pouvant opérer en temps réel.

Des simulations sur la supervision d'un système de transport intelligent dans le cadre du projet InTraDE ont été réalisées à partir d'un modèle d'hypergraphe sur une plateforme de simulation en temps réel s'agissant de SCANER studio [OKTAL].

CHAPITRE 4

Modélisation des systèmes de systèmes par l'approche du graphe dynamique

Sommaire

4.1	Introduction	88
4.1.1	Contexte scientifique	88
4.1.2	Contexte applicatif	89
4.2	Passage du modèle hypergraphe à un modèle de graphe dynamique	90
4.3	Modèle de graphe dynamique d'un système de systèmes	92
4.4	Optimisation en ligne sur un modèle de graphe dynamique	94
4.4.1	Extraction du graphe statique \mathfrak{N}_t	95
4.4.2	Décomposition de \mathfrak{N}_t et \mathcal{G}_t en sous-graphes	96
4.4.3	Valuation des arcs du graphe statique \mathfrak{N}_t	98
4.5	Application à un système de transport intelligent	98
4.6	Résultats de Simulation	105
4.6.1	Description du terrain	105
4.6.2	Scénario de routage	105
4.7	Conclusion	112

4.1 Introduction

4.1.1 Contexte scientifique

Dans le chapitre 3, nous avons montré comment un Système de Systèmes (SdS) peut être représenté graphiquement par un hypergraphe. Le modèle obtenu regroupe les composants systèmes des différents niveaux grâce à des hyperarêtes, dont la propriété est de lier plusieurs nœuds entre eux. Ce modèle d'hypergraphe décrit les composants systèmes du SdS sur plusieurs niveaux, permettant ainsi d'élaborer une stratégie de supervision.

Nous avons aussi montré, via les propriétés graphiques de l'hypergraphe, que la reconfiguration du SdS est possible en parcourant cet hypergraphe de façon descendante, à travers une stratégie d'optimisation verticale dite à multi-niveaux.

Lorsque nous souhaitons élaborer une stratégie d'optimisation transversale correspond à un seul niveau de modélisation d'un SdS, telle que l'optimisation de trafic dans un système de transport intelligent (ITS), il n'est souvent pas utile de considérer tous les composants du SdS répartis sur l'ensemble des niveaux. En effet, dans le cas où un seul niveau de modélisation est considéré, seuls les composants systèmes appartenant à ce niveau ainsi que leurs interactions sont à prendre en compte.

Pour résoudre ce problème d'optimisation, dit à un niveau, nous avons besoin d'extraire du modèle multi-niveaux de l'hypergraphe un seul niveau obtenu en découplant les liens entre composants systèmes de ce niveau et composants systèmes des autres niveaux. Le résultat généré est un Modèle de Graphe Dynamique (MGD) dont les éléments (nœuds, arcs et poids) sont dynamiques.

La théorie des graphes a montré des performances significatives dans la modélisation des opérations de transport. En utilisant la représentation graphique, le chemin optimal entre les nœuds statiques ou dynamiques peut être facilement déduit en utilisant peu de développement et de calcul. Dans la littérature, deux types de modèles graphiques existent : les graphes statiques et les graphes dynamiques.

Le graphe statique [VLIET, 1978], [GALLO, 1986], [AHUJA, 1990] est indépendant du temps par contre le graphe dynamique [PALLOTTINO, 1997], [FERREIRA, 2002], [CORTES, 2003], [DEMETRESCU, 2006] est structurellement modifiable et dépend du temps. Généralement, la modification struc-

turelle du graphe dynamique réside dans la suppression ou l'ajout de nœuds ou d'arcs.

Dans ce chapitre, un modèle de graphe dynamique du SdS est proposé, extrait du modèle hypergraphe développé dans le chapitre précédent, et appliqué à un système ITS. Ce dernier est composé de véhicules autonomes intelligents (IAVs) opérant dans un espace confiné incluant une infrastructure routière et un environnement dynamique. Ce modèle permet d'implémenter des algorithmes d'optimisation pouvant améliorer la gestion du trafic en utilisant les IAVs tout en s'adaptant aux changements dynamiques des composants systèmes du système ITS. L'algorithme d'optimisation proposé à partir de ce graphe permet d'estimer de manière adaptative le temps qu'un IAV prendra pour parcourir une distance entre deux nœuds (statique ou dynamique) en fonction de l'état des différents composants systèmes du système ITS.

4.1.2 Contexte applicatif

Un système ITS peut être considéré comme un SdS. Il satisfait ses principales propriétés, à savoir, l'indépendance opérationnelle, l'indépendance managériale, la dispersion géographique, le comportement émergent et coopératif et le développement évolutionnaire [KRYGIEL, 1999], [CARLOCK, 2001]. Dans le cas du transport routier, nous avons vu qu'un système ITS associe aux systèmes de transport des systèmes ICT afin d'améliorer la sécurité et l'efficacité du transport. Ces technologies permettent d'échanger l'information entre les véhicules (véhicule à véhicule : V2V) et entre les véhicules et son environnement (véhicule à l'infrastructure : V2I). Cependant, un système ITS ne se limite pas au transport routier, il est aussi présent dans d'autres types de transport : ferroviaire, aérien [MUKHERJEE, 2004] et maritime [REVIEW].

Les IAVs sont des composants systèmes d'un système ITS, qui opèrent souvent dans des espaces confinés et sécurisés. Ces véhicules sont utilisés pour améliorer la gestion du trafic, par exemple la vitesse du véhicule est adaptée en fonction de l'état du trafic. De nos jours, des véhicules guidés et automatisés (AGVs) sont exploités dans plusieurs espaces tels que : ports, aéroports et zones logistiques. Ces véhicules sont caractérisés par le suivi guidé sur le terrain en utilisant des systèmes ICT, où l'environnement d'exploitation doit s'adapter généralement au fonctionnement des AGVs. L'avantage d'utiliser les IAVs dans des applications de routage réside dans leur adaptation à des environnements variables contrairement aux AGVs.

L'exploitation des IAVs dans des espaces confinés de type ports maritimes contribue à améliorer la gestion du trafic interne, à travers une gestion optimale du routage. Cette problématique d'optimisation est complexe car elle dépend de plusieurs paramètres tels que la cartographie du terrain, la densité de la circulation, la météo, l'état des véhicules Ainsi, l'optimisation d'un tel système de transport nécessite l'élaboration d'un modèle dynamique adapté aux opérations du trafic dans un environnement complexe.

4.2 Passage du modèle hypergraphe à un modèle de graphe dynamique

Un modèle hypergraphe représente l'aspect hiérarchique (multi-niveaux) d'un SdS (FIGURE 4.1). Cette représentation hiérarchique est adéquate pour la recherche d'une configuration opérationnelle du SdS, obtenue par exemple, à partir de l'analyse des possibilités de reconfiguration, comme expliqué dans le chapitre 3.

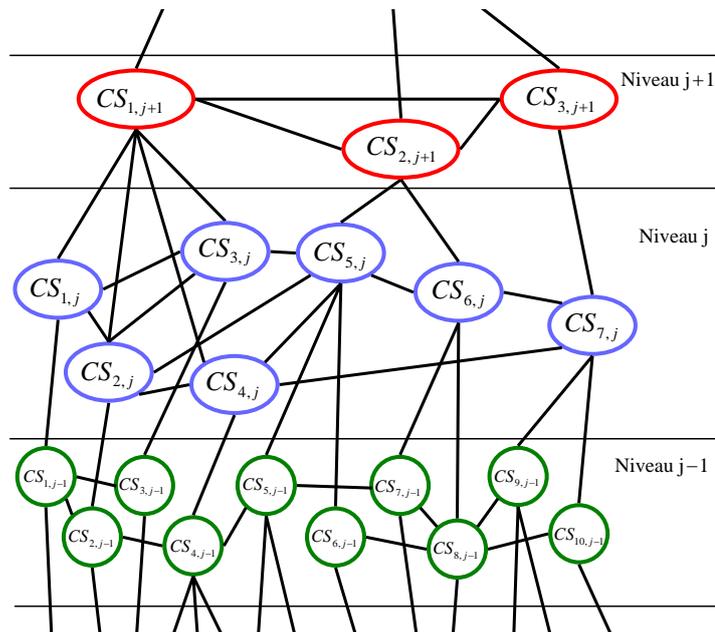


FIGURE 4.1 – Modèle multi-niveaux d'un SdS

Cette recherche est basée sur la résolution d'un problème d'optimisation verticale ou multi-niveaux. Cependant, dans certains cas, il est utile de reconfigurer localement un SdS sur un niveau de modélisation en fonction du mode

de fonctionnement des composants systèmes de ce niveau. Le problème d'optimisation dit transversal se réduit alors à un seul niveau de modélisation.

Cette section montre la méthodologie graphique pour extraire un niveau de modélisation à partir d'un modèle hiérarchique multi-niveaux représenté par un hypergraphe. Puisque le SdS est représenté par un hypergraphe imbriqué et valué, et afin de ne pas perdre les informations échangées ($ie_{i,j}$) entre les composants systèmes de différents niveaux lors du passage à un niveau de modélisation, ces dernières doivent être considérées comme des entrées dans la nouvelle représentation de modélisation à un niveau (FIGURE 4.2).

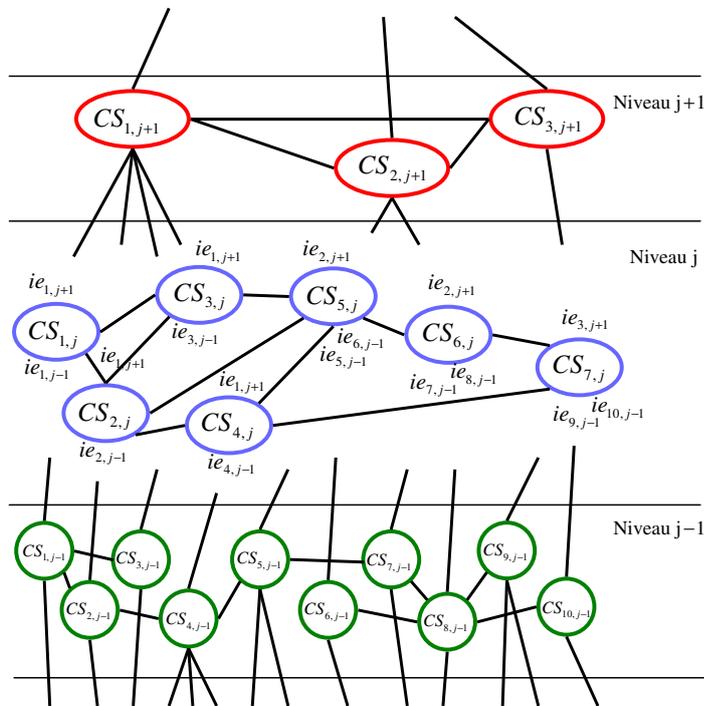


FIGURE 4.2 – Découplage du niveau j par rapport aux niveaux $j - 1$ et $j + 1$

En effet, chaque niveau de l'hypergraphe contient un certain nombre de composants systèmes représentés par des hyperarêtes interconnectées. Cette représentation est due au fait que chaque composant système peut contenir plusieurs autres composants systèmes de niveaux inférieurs. D'un point de vue local, nous ne nous intéressons pas aux composants systèmes de niveaux inférieurs et/ou supérieurs, mais uniquement à des composants systèmes d'un seul niveau de modélisation. Par conséquent, pour ce niveau, les hyper-

arêtes sont considérées comme des nœuds, les liens entre les hyperarêtes sont considérés comme des arcs et les informations dynamiques échangées avec les autres niveaux sont considérées comme des poids sur les nœuds (FIGURE 4.3). Avec ces caractéristiques graphiques, nous obtenons un MGD représentant un seul niveau du modèle hypergraphe.

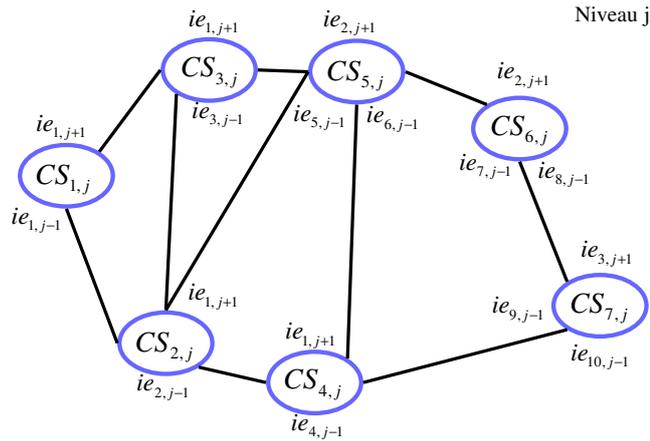


FIGURE 4.3 – Modèle de graphe dynamique du niveau j

4.3 Modèle de graphe dynamique d'un système de systèmes

Le graphe dynamique se caractérise par sa prise en compte du temps et à sa structure évolutionnaire. Ces deux caractéristiques introduisent deux aspects pour le graphe dynamique : local et global. L'aspect local décrit l'évolution temporelle de la valuation des éléments du graphe (nœuds ou arcs), l'aspect global rend possible l'ajout ou la suppression d'éléments. On notera que la modification des valeurs des éléments graphiques peut provoquer des changements structuraux du graphe global. De même, ajouter ou supprimer des éléments, peut changer les valeurs des éléments non modifiés du graphe.

Dans cette section, nous présentons un modèle de graphe dynamique pour un SdS qui prend en compte les deux aspects local et global.

Considérons un SdS composé de composants systèmes statiques et dynamiques représentés par des nœuds (FIGURE 4.4). Un nœud statique est géographiquement fixe et correspond à un nœud source ou à un nœud desti-

Définition 4.2. L'ensemble $\mathcal{A} = \{(N_i, N_j) / N_i, N_j \in \mathcal{N}, i \neq j\}$ est un ensemble fini d'arcs orientés. Chaque arc de $\mathcal{A}_{i,j}$ connecte N_i à N_j . Il est représenté par le couple (N_i, N_j) . Ces arcs sont orientés, où l'ordre de N_i et N_j est important dans le couple (N_i, N_j) , avec $i, j \in \mathbb{N}^*$ et $1 \leq i, j \leq nb_N$.

Définition 4.3. L'ensemble $\mathfrak{F}_G = \{f_i / 0 \leq i \leq nb_F\}$ est un ensemble de fonctions temporelles, associé au graphe \mathcal{G}_t .

Définition 4.4. D'un point de vue local, l'évolution temporelle de la valuation des éléments graphiques est représentée par des fonctions f_i , où f_i est définie par :

$$f_i : \begin{array}{ll} (\mathcal{N} \cup \mathcal{A}) \times \mathbb{R}^+ & \rightarrow E_{f_i} \\ (X, t) & \rightarrow f_i(X, t), \end{array}$$

avec E_{f_i} l'ensemble des valeurs associées à la fonction f_i , et X un nœud ou un arc.

On peut associer par exemple des fonctions aux nœuds pour indiquer le mode de fonctionnement des composants systèmes. D'autres fonctions peuvent être associées aux arcs pour montrer des relations entre les composants systèmes comme la distance ou autres types d'informations échangées.

Définition 4.5. D'un point de vue global, l'ajout d'un nouveau nœud N_i conduit à l'apparition de l'ensemble $\mathcal{N} \cup \{N_i\}$, alors que la suppression d'un nœud existant provoque à la fois l'apparence de $\mathcal{N} \setminus \{N_i\}$ et

$$\mathcal{A} \setminus \bigcup_{j,k \in \{1, \dots, n\}} \{(N_j, N_k) \in \mathcal{N} \times \mathcal{N}; N_j = N_i \text{ ou } N_k = N_i\}.$$

L'ajout d'un nouvel arc ou la suppression d'un arc existant provoque respectivement l'apparition de $\mathcal{A} \cup \{(N_i, N_j)\}$ ou $\mathcal{A} \setminus \{(N_i, N_j)\}$.

4.4 Optimisation en ligne sur un modèle de graphe dynamique

Prenons le cas d'un graphe dynamique de SdS appliqué au système ITS. Un composant système est dit dynamique, lorsqu'il est représenté par un nœud dynamique¹ constamment mobile entre deux nœuds statiques².

-
1. dispersion géographique et évolutionnaire
 2. composants systèmes statiques

Dans la littérature, plusieurs travaux sur l'optimisation des graphes dynamiques existent [PALLOTTINO, 1997], [FERREIRA, 2002], [CORTES, 2003], [DEMETRESCU, 2006]. Ces derniers ne considèrent pas à la fois les aspects local et global du MGD. D'autres travaux, [BELLMAN, 1958], [DIJKSTRA, 1959], [VLIET, 1978], [AHUJA, 1990], portent sur le développement d'algorithmes pour la recherche du chemin optimal à un instant donné sur un graphe statique.

L'optimisation appliquée sur un graphe dynamique ne peut être qu'une optimisation en ligne [FRIESE, 2006], [PALMIERI, 2010], [MAHR, 2010], s'adaptant à l'évolution du graphe. Cette représentation du graphe dynamique peut être expliquée par une évolution saccadée de plusieurs graphes statiques. Ainsi, une solution optimale pour le déplacement d'un nœud dynamique dans un MGD, consiste à calculer en permanence le chemin à parcourir instantanément par ce nœud, de telle façon que la somme des valeurs des arcs du MGD à cet instant soit minimisée [CORMEN, 2001].

La stratégie d'optimisation développée dans cette section est appliquée sur un MGD décrivant à la fois l'aspect local (temps) et global (évolution). Elle s'inspire de la décomposition du graphe dynamique en une succession continue de graphes statiques sur lesquels nous pouvons appliquer différents algorithmes de calcul du chemin optimal.

Cette stratégie permet de trouver de façon adaptative un chemin optimal entre un nœud source et un nœud destination à partir d'un MGD. Elle est résumée par les étapes suivantes :

1. extraction du graphe statique \mathfrak{N}_t à partir du graphe dynamique \mathcal{G}_t ;
2. décomposition des graphes statique et dynamique \mathcal{G}_t et \mathfrak{N}_t en sous-graphes ;
3. valuation des arcs du graphe statique \mathfrak{N}_t .

Une fois le modèle conçu à partir du graphe statique valué, un algorithme d'optimisation approprié est ainsi élaboré.

4.4.1 Extraction du graphe statique \mathfrak{N}_t

Considérons le graphe statique $\mathfrak{N}_t(\mathcal{N}_{\mathfrak{N}}, \mathcal{R}, \mathfrak{F}_{\mathfrak{N}})$ issu du graphe \mathcal{G}_t (FIGURE 4.5), où $\mathcal{N}_{\mathfrak{N}} = N_S \cup \{\alpha, \beta\}$ est l'ensemble des nœuds statiques N_S à l'instant t , dont le nœud source est α , et le nœud destination est β . $\mathcal{R} = \{R_{i,j} = (N_i, N_j)/N_i, N_j \in \mathcal{N}_{\mathfrak{N}}, i \neq j\}$ est un ensemble fini d'arcs. Enfin, $\mathfrak{F}_{\mathfrak{N}}$ est un

ensemble fini de fonctions temporelles, associées au graphe \mathfrak{N}_t . Le graphe statique \mathfrak{N}_t est ainsi obtenu après le retrait de l'ensemble des nœuds dynamiques du graphe \mathcal{G}_t .

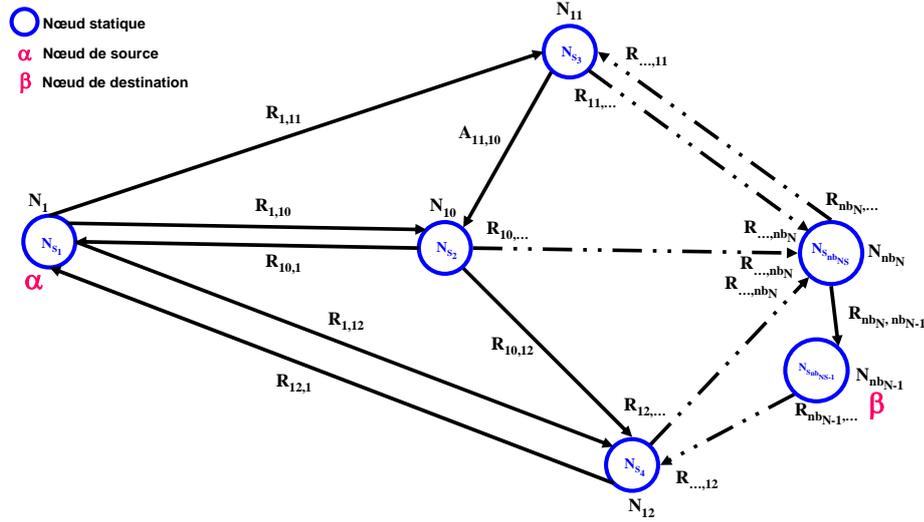


FIGURE 4.5 – Extraction du graphe statique \mathfrak{N}_t à partir du graphe dynamique \mathcal{G}_t de la FIGURE 4.4

4.4.2 Décomposition de \mathfrak{N}_t et \mathcal{G}_t en sous-graphes

1. La décomposition du graphe statique \mathfrak{N}_t de la FIGURE 4.5 en sous-graphes \mathcal{SN} consiste à dissocier toutes les branches unidirectionnelles situées entre deux nœuds voisins N_i et N_j (FIGURE 4.6).
2. Par analogie, la décomposition du graphe dynamique \mathcal{G}_t de la FIGURE 4.4 en sous-graphes \mathcal{SG} se fait par l'extraction des branches unidirectionnelles situées entre deux nœuds N_i et N_j incluant tous les nœuds intermédiaires (FIGURE 4.7).

Pour chaque sous-graphe $\mathcal{SN}_{i,j}$ nous associons un unique sous-graphe $\mathcal{SG}_{i,j}$, représenté par une bijection \mathfrak{B} définie comme suit :

$$\mathfrak{B} : \begin{array}{l} \mathfrak{N}_t \rightarrow \mathcal{G}_t \\ \mathcal{SN}_{i,j} \rightarrow \mathfrak{B}(\mathcal{SN}_{i,j}) = \mathcal{SG}_{i,j}. \end{array}$$

4.4.3 Valuation des arcs du graphe statique \mathfrak{N}_t

Après avoir défini le graphe statique \mathfrak{N}_t , les arcs peuvent maintenant être évalués de façon dynamique par une fonction \mathfrak{w} prenant en considération le comportement des nœuds dynamiques du graphe \mathcal{G}_t selon le cas à étudier :

$$\forall R_{i,j} \in \mathcal{R}, w_{i,j}(t) = \mathfrak{w}(\mathcal{G}_t(\mathcal{N}, \mathcal{A}, \mathfrak{F}_{\mathcal{G}})).$$

4.5 Application à un système de transport intelligent

Considérons le SdS représenté par un système ITS, contenant des stations (\mathcal{S}), des jonctions (\mathcal{J}), des clients (\mathcal{C}), et des IAVs (\mathcal{V}). Une station est l'endroit où des conteneurs transitent de façon bidirectionnelle à l'aide des véhicules IAVs. Une jonction est associée à une intersection. Initialement, chaque IAV peut accéder à tous les chemins et peut changer sa trajectoire durant sa mission. Nous considérons que le terrain des zones de routage est bien identifié. Chaque IAV est considéré comme un composant système dynamique de l'ensemble du système.

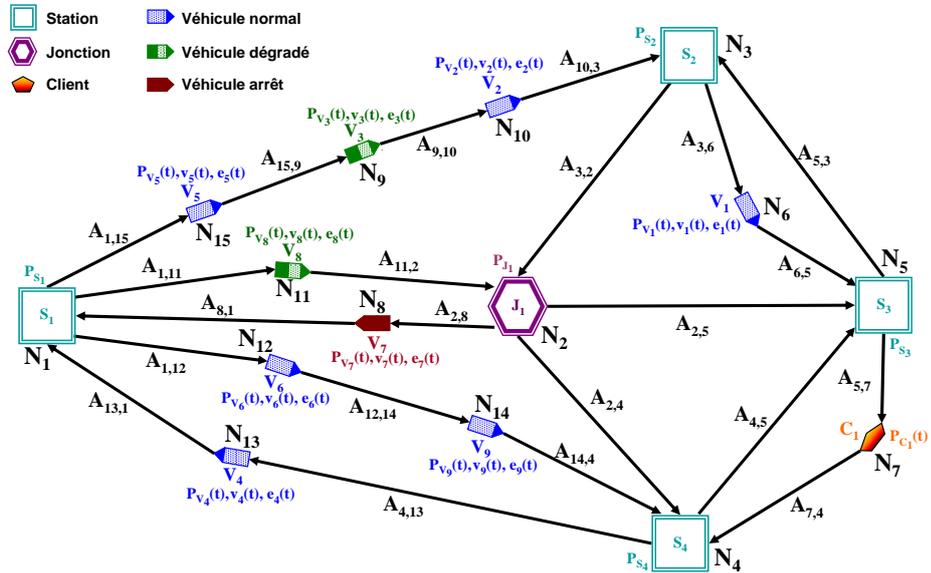


FIGURE 4.8 – Graphe dynamique \mathcal{G}_t du système ITS

Le modèle dynamique du système ITS noté \mathcal{G}_t peut être représenté graphiquement comme proposé par la FIGURE 4.8. Les nœuds statiques (N_S)

représentent les stations (\mathcal{S}) et les jonctions (\mathcal{J}), $N_S = \mathcal{S} \cup \mathcal{J}$, où $\mathcal{S} = \{S_i / 1 \leq i \leq nb_S\}$, $\mathcal{J} = \{J_i / 1 \leq i \leq nb_J\}$, $nb_{N_S} = nb_S + nb_J$ et $\mathcal{S} \cap \mathcal{J} = \emptyset$. Alors que les nœuds dynamiques (N_D) représentent des clients (\mathcal{C}) et les IAVs (\mathcal{V}), $N_D = \mathcal{C} \cup \mathcal{V}$, où $\mathcal{C} = \{C_i / 1 \leq i \leq nb_C\}$, $\mathcal{V} = \{V_i / 1 \leq i \leq nb_V\}$, $nb_{N_D} = nb_C + nb_V$ et $\mathcal{C} \cap \mathcal{V} = \emptyset$. Où nb_{N_S} , nb_S , nb_J , nb_{N_D} , nb_C , $nb_V \in \mathbb{N}$, est respectivement le nombre de nœuds statiques, de stations, de jonctions, de nœuds dynamiques, de clients et de véhicules IAVs.

L'ensemble des fonctions temporelles associées au système peut être décrit comme suit :

$$\tilde{\mathfrak{F}}_G = \{\mathfrak{p}_S, \mathfrak{p}_J, \mathfrak{p}_C, \mathfrak{p}_V, \mathfrak{v}, \mathfrak{e}\}$$

► \mathfrak{p}_S et \mathfrak{p}_J sont deux fonctions constantes associées à \mathcal{S} et \mathcal{J} respectivement, où $i, j \in \mathbb{N}^*$, $1 \leq i \leq nb_S$ et $1 \leq j \leq nb_J$:

$$\begin{aligned} \mathfrak{p}_S : \mathcal{S} &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \\ S_i &\rightarrow \mathfrak{p}_S(S_i) = P_{S_i} = (x_{S_i}, y_{S_i}, z_{S_i}) \end{aligned}$$

$$\begin{aligned} \mathfrak{p}_J : \mathcal{J} &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \\ J_j &\rightarrow \mathfrak{p}_J(J_j) = P_{J_j} = (x_{J_j}, y_{J_j}, z_{J_j}) \end{aligned}$$

P_{S_i} et P_{J_j} représentent respectivement la position statique et absolue de la $i^{\text{ème}}$ station et la $j^{\text{ème}}$ jonction.

► \mathfrak{p}_C et \mathfrak{p}_V sont deux fonctions temporelles associées aux \mathcal{C} et \mathcal{V} respectivement, où $i, j \in \mathbb{N}^*$, $1 \leq i \leq nb_C$ et $1 \leq j \leq nb_V$:

$$\begin{aligned} \mathfrak{p}_C : \mathcal{C} \times \mathbb{R}^+ &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \\ (C_i, t) &\rightarrow \mathfrak{p}_C(C_i, t) = P_{C_i}(t) = (x_{C_i}(t), y_{C_i}(t), z_{C_i}(t)) \end{aligned}$$

$$\begin{aligned} \mathfrak{p}_V : \mathcal{V} \times \mathbb{R}^+ &\rightarrow \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \\ (V_j, t) &\rightarrow \mathfrak{p}_V(V_j, t) = P_{V_j}(t) = (x_{V_j}(t), y_{V_j}(t), z_{V_j}(t)) \end{aligned}$$

$P_{C_i}(t)$ et $P_{V_j}(t)$ représentent respectivement la position dynamique du $i^{\text{ème}}$ client et $j^{\text{ème}}$ IAV à l'instant t .

► \mathfrak{v} est une fonction temporelle associée à \mathcal{V} , où $i \in \mathbb{N}^*$ et $1 \leq i \leq nb_V$, telle que :

$$\begin{aligned} \mathfrak{v} : \mathcal{V} \times \mathbb{R}^+ &\rightarrow \mathbb{R}^+ \\ (V_i, t) &\rightarrow \mathfrak{v}(V_i, t) = v_i(t) \end{aligned}$$

$v_i(t)$ représente la vitesse du $i^{\text{ème}}$ IAV à l'instant t .

► \mathbf{e} est une fonction temporelle associée à \mathcal{V} , où $i \in \mathbb{N}^*$ et $1 \leq i \leq nb_V$:

$$\begin{aligned} \mathbf{e} : \mathcal{V} \times \mathbb{R}^+ &\rightarrow \{V_N, V_F, V_D\} \\ (V_i, t) &\rightarrow \mathbf{e}(V_i, t) = e_i(t) \end{aligned}$$

$e_i(t)$ indique l'état du $i^{\text{ème}}$ IAV à l'instant t .

Le graphe statique $\mathfrak{N}_t(N_{\mathfrak{N}}, \mathcal{R}, \mathfrak{F}_{\mathfrak{N}})$ extrait du graphe dynamique \mathcal{G}_t est utilisé pour trouver le chemin optimal que doit emprunter un IAV en présence de n autres IAVs. Les arcs $R_{i,j}$ (routes dans le cas d'un ITS) sont supposés unidirectionnels, ce qui signifie qu'une route avec une double circulation est représentée par deux arcs opposés $R_{i,j}$ et $R_{j,i}$ (FIGURE 4.9).

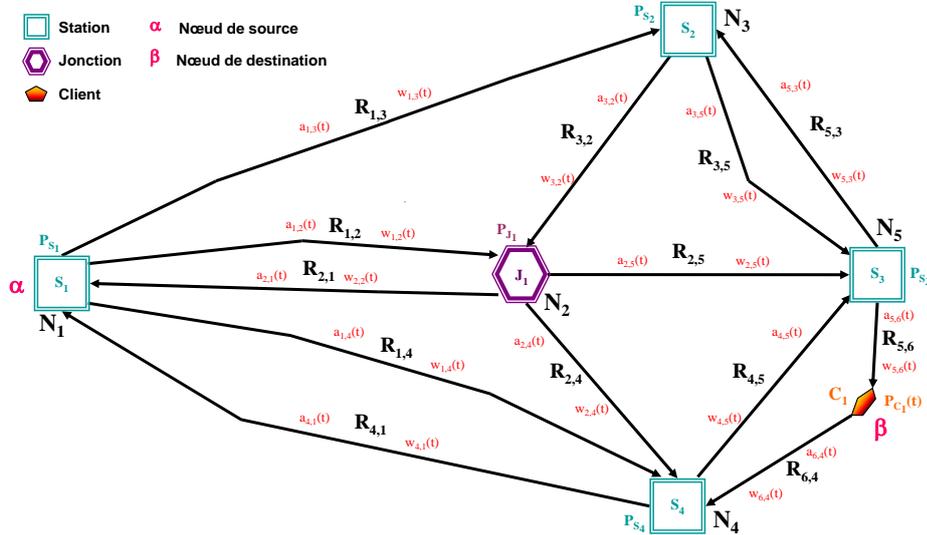


FIGURE 4.9 – Le graphe statique \mathfrak{N}_t

L'ensemble des fonctions temporelles associées au ITS est défini comme suit :

$$\mathfrak{F}_{\mathfrak{N}} = \{\mathbf{p}_S, \mathbf{p}_{\mathcal{J}}, \mathbf{p}_\alpha, \mathbf{p}_\beta, \mathbf{a}, \mathbf{w}\}.$$

► \mathbf{p}_α et \mathbf{p}_β sont deux fonctions temporelles associées respectivement au nœud source α et au nœud destination β dans le cas dynamique.

► \mathbf{a} est une fonction booléenne associée à \mathcal{R} , où $i, j \in \mathbb{N}^*$, $i \neq j$ et $1 \leq i, j \leq nb_{N_{\mathfrak{N}}}$:

$$\begin{aligned} \mathbf{a} : \mathcal{R} \times \mathbb{R}^+ &\rightarrow \{0, 1\} \\ (R_{i,j}, t) &\rightarrow \mathbf{a}(R_{i,j}, t) = a_{i,j}(t) \end{aligned}$$

$a_{i,j}(t)$ représente l'accessibilité à la route $R_{i,j}$ à l'instant t . Sa valeur est égale à 1 si la route $R_{i,j}$ est accessible et 0 sinon.

► \mathfrak{w} est une fonction temporelle associée à \mathcal{R} , où $i, j \in \mathbb{N}^*$, $i \neq j$, $1 \leq i, j \leq nb_{N_{\mathfrak{N}}}$:

$$\begin{aligned} \mathfrak{w} : \mathcal{R} \times \mathbb{R}^+ &\rightarrow \mathbb{R}^+ \\ (R_{i,j}, t) &\rightarrow \mathfrak{w}(R_{i,j}, t) = w_{i,j}(t) \end{aligned}$$

$w_{i,j}(t)$ représente le temps relatif nécessaire pour parcourir la route $R_{i,j}$ à l'instant t .

La fonction \mathfrak{w} peut être définie par l'algorithme de valuation suivant :

Algo. 1 : Algorithme de valuation

Entrée

$\mathcal{G}_t(\mathcal{N}, \mathcal{A}, \mathfrak{F}_{\mathcal{G}})$, $\mathfrak{N}_t(N_{\mathfrak{N}}, \mathcal{R}, \mathfrak{F}_{\mathfrak{N}} \setminus \{\mathfrak{w}\})$;

Sortie

\mathfrak{w} .

Début

pour chaque route $R_{i,j}$ dans \mathcal{R}

$w_{i,j}(t) = 0$;

Si il n'y a aucun véhicule sur $R_{i,j}$ **alors**

$w_{i,j}(t) = t_{i,j}$;

Sinon Si $e_k(t) = V_N$, $\forall k \leq n$ **alors**

$w_{i,j}(t) = t_{i,j}$;

Sinon Si $\exists k \leq n$ / $e_k(t) = V_A$ **alors**

$w_{i,j}(t) = \infty$;

Sinon

Répéter

Pour chaque $k \leq n$

Si $v_k(t) < v_{k+1}(t)$ **alors**

Si $\tilde{d}_{i,V_{k+1}} \leq d_{i,j}$ **alors**

calculer $\tilde{t}_{i,V_{k+1}}$;

localiser les $(n+1)$ véhicules selon $\min(\tilde{t}_{i,V_{k+1}})$;

$w_{i,j}(t) = w_{i,j}(t) + \min(\tilde{t}_{i,V_{k+1}})$;

Jusqu'à que $\tilde{d}_{i,V_{k+1}} > d_{i,j}$, $\forall k \neq n$

$w_{i,j}(t) = w_{i,j}(t) + t_{V_{n+1},j}$;

Fin

Pour appliquer cet algorithme, nous considérons les hypothèses suivantes :

1. la zone d'expérimentation est considérée confinée (limitée) et son réseau routier est bien identifié,
2. le nombre ainsi que l'état des véhicules IAVs et les stations sont parfaitement connus à l'instant t ,
3. un IAV ne peut pas dépasser son précédent. Il adapte en continu sa vitesse,
4. le temps d'attente au niveau des jonctions est négligeable, à cause du nombre limité des IAVs engagés,
5. les IAVs opèrent de façon autonome et automatique selon les consignes du système de supervision du système ITS,
6. les IAVs se déplacent sur une route avec un mouvement rectiligne uniforme.

On note :

- n : le nombre de véhicules engagés sur l'arc $R_{i,j}$,
- $d_{i,j}$: la distance entre le nœud N_i et le nœud N_j ,
- $d_{i,V_k}(t)$: la distance temporelle entre le nœud N_i et le véhicule V_k ,
- $\tilde{d}_{i,V_{k+1}}(t)$: la distance temporelle entre N_i et la position de sécurité que le véhicule V_k doit garder par rapport au véhicule V_{k+1} ,
- $t_{i,j}$: le temps nécessaire pour parcourir l'arc $R_{i,j}$ dans des conditions normales,
- $\tilde{t}_{i,V_{k+1}}$: le temps nécessaire pour parcourir $(\tilde{d}_{i,V_{k+1}}(t) - d_{i,V_{k+1}}(t))$,
- $\tilde{t}_{V_{n+1},j}$: le temps nécessaire pour parcourir $d_{V_{n+1},j}(t)$,
- $\min(\tilde{t}_{i,V_{k+1}})$: la valeur minimale de $\tilde{t}_{i,V_{k+1}}$, avec $1 \leq k \leq n$.

Soit $d_{i,j} = \sqrt{(x_{N_j} - x_{N_i})^2 + (y_{N_j} - y_{N_i})^2 + (z_{N_j} - z_{N_i})^2}$, la longueur linéaire de l'arc (route) $R_{i,j}$.

L'algorithme de valuation (Algo. 1) explique comment on peut estimer le temps relatif ($w_{i,j}(t)$) utile permettant au $(n + 1)^{\text{ème}}$ IAV de parcourir une route $R_{i,j}$, contenant n IAVs, avec $n \geq 0$:

► Si $n = 0$, il n'y a aucun IAV sur la route $R_{i,j}$, alors le temps relatif estimé ($w_{i,j}(t)$) est égal à $t_{i,j}$.

► Si $n > 0$, on distingue trois cas :

i) Si les n IAVs sur la route $R_{i,j}$ sont en mode de fonctionnement normal

en absence d'obstacle, alors le temps relatif ($w_{i,j}(t)$) est égal à la durée initiale calculée pour parcourir la route $R_{i,j}$ où la vitesse désirée (FIGURE 4.10-a) :

$$w_{i,j}(t) = t_{i,j} = d_{i,j}/v_{i,j}.$$

ii) Si un IAV est endommagé et arrêté, la route $R_{i,j}$ devient non accessible ($a_{i,j}(t) = 0$), et le temps relatif ($w_{i,j}(t)$) est égal à l'infini. Ce cas oblige le système de supervision de ITS à générer un nouveau scénario de routage (FIGURE 4.10-b).

iii) Le troisième cas représente le cas où un ou plusieurs véhicules sont en mode dégradé parmi les IAVs engagés (FIGURE 4.10-c). Le $(n + 1)^{\text{ème}}$ IAV peut atteindre la destination (N_j) après l'arrivée du $n^{\text{ème}}$ IAV et ainsi de suite. L'estimation du temps relatif ($w_{i,j}(t)$) pour que le $(n + 1)^{\text{ème}}$ IAV puisse atteindre sa destination, est déduite de l'estimation de tous les temps de trajet $R_{i,j}$ effectués par les k IAVs pour tous $k \in \{1, \dots, n\}$.

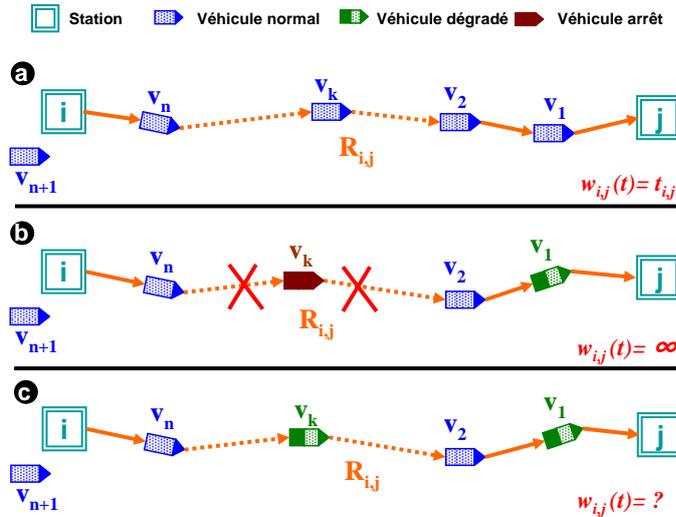


FIGURE 4.10 – Différents scénarios de routage des IAVs sur une route $R_{i,j}$

Pour estimer ce temps, nous commençons par comparer la vitesse relative des deux premiers IAVs (V_1 et V_2) sur la route $R_{i,j}$. Lorsque la vitesse du premier est inférieure à la vitesse du suiveur, nous estimons la distance \tilde{d}_{i,V_2} , (FIGURE 4.11). Selon la valeur de \tilde{d}_{i,V_2} , nous distinguons deux cas :

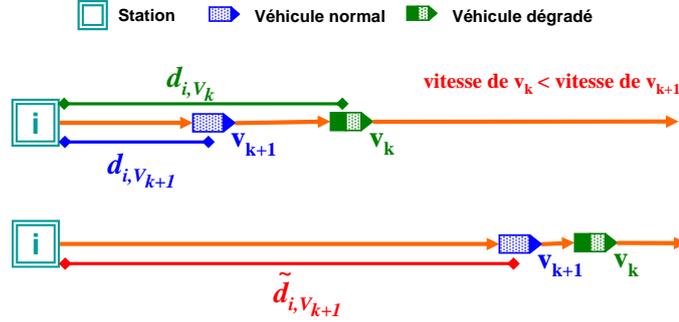


FIGURE 4.11 – La distance $\tilde{d}_{i,v_{k+1}}$

- ▷ Si \tilde{d}_{i,v_2} est plus grande ou égale à $d_{i,j}$ (FIGURE 4.12-a), alors le premier IAV atteint le nœud N_j avant son suiveur.
- ▷ Si \tilde{d}_{i,v_2} est inférieure à $d_{i,j}$, le véhicule suiveur peut attendre son prédécesseur après \tilde{t}_{i,v_2} (FIGURE 4.12-b).

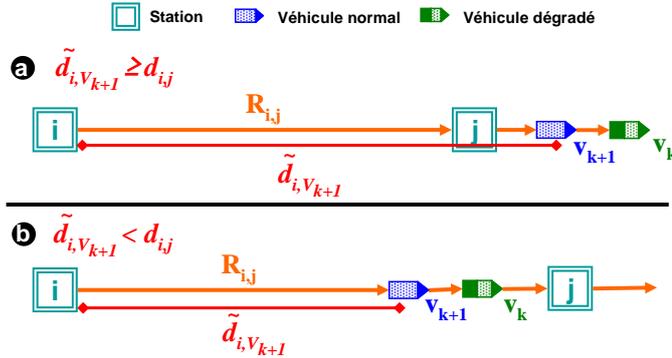


FIGURE 4.12 – Les deux cas en fonction de $\tilde{d}_{i,v_{k+1}}$

De la même façon, nous estimons les autres distances $\tilde{d}_{i,v_3}, \dots, \tilde{d}_{i,v_{n+1}}$. Ensuite, nous localisons les $n + 1$ IAVs sur la route $R_{i,j}$ en fonction de $\min(\tilde{t}_{i,v_{k+1}})$. Nous répétons cette étape jusqu'à ce que nous obtenions toutes les distances \tilde{d}_{i,v_k} supérieures ou égale à $d_{i,j}$. Le temps relatif estimé pour que le $(n + 1)^{\text{ème}}$ IAV puisse parcourir la route spécifiée $R_{i,j}$ peut ensuite être calculé.

Après l'estimation de l'ensemble des valeurs $w_{i,j}(t)$, le graphe statique \mathfrak{N}_t est finalement valué. Dans la section suivante, un scénario simulé montre comment nous pouvons appliquer l'algorithme de [DIJKSTRA, 1959] sur le

graphe développé \mathfrak{N}_t , afin de trouver le chemin optimal, entre une source et une destination.

4.6 Résultats de Simulation

Le scénario de simulation est réalisé sur le logiciel SCANer studio [OKTAL]. Ce dernier est un simulateur opérant en temps réel, dédié à la recherche et au développement.

4.6.1 Description du terrain

Nous appliquons le MGD sur une zone portuaire du terminal « Radicatel » en Normandie (France), dont la cartographie est montrée FIGURE 4.13. Nous représentons le plan comme un graphe orienté valué afin de trouver le chemin optimal, en prenant en considération le nombre et l'état des IAVs qui se déplacent entre les zones d'importation et d'exportation.

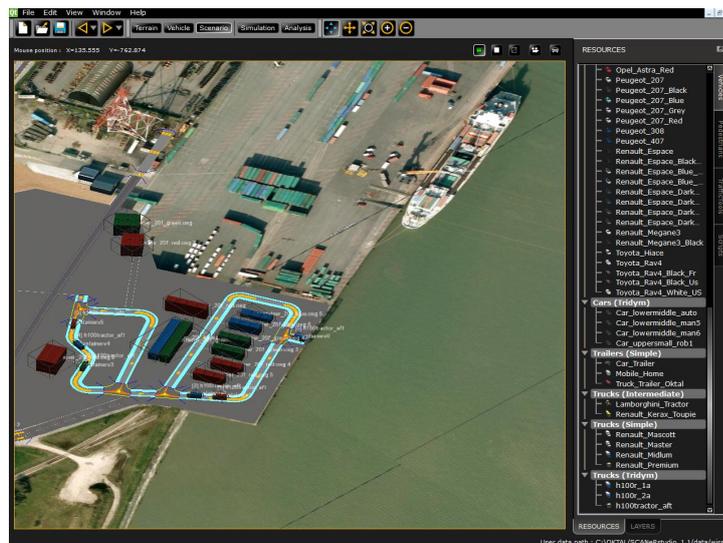


FIGURE 4.13 – Terminal maritime « Radicatel » sous SCANer studio

4.6.2 Scénario de routage

Soit le graphe de la FIGURE 4.14, représentant la partie considérée du port où les opérations vont être simulées. Comme nous pouvons le voir, il existe plusieurs parcours possibles entre la source S_1 et la destination S_2 à savoir :

1. $S_1; J_3; J_4; J_5; J_6; J_{10}; J_{11}; S_2$
2. $S_1; J_1; J_2; J_4; J_5; J_6; J_{10}; J_{11}; S_2$
3. $S_1; J_3; J_4; J_5; J_7; J_8; J_9; J_{10}; J_{11}; S_2$
4. $S_1; J_1; J_2; J_4; J_5; J_7; J_8; J_9; J_{10}; J_{11}; S_2$

Pour ce scénario de routage, nous étudions trois possibilités. La première concerne le plus court parcours ; la seconde consiste à choisir le parcours avec le moins de véhicules engagés, tandis que la troisième applique l'algorithme d'estimation développé dans ce travail.

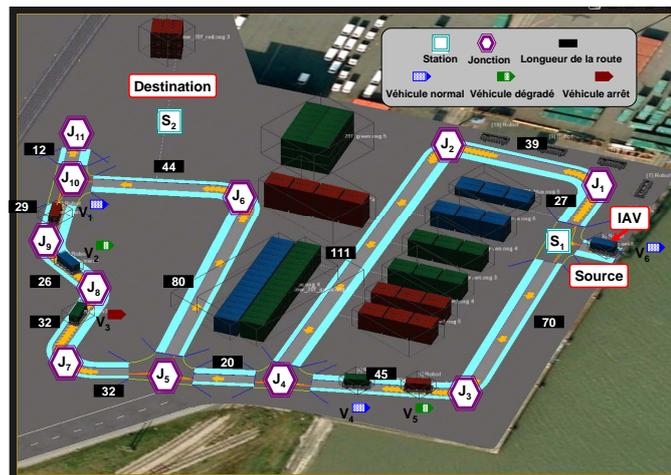


FIGURE 4.14 – Opération de routage dans le domaine de l'exportation

Les longueurs des routes du réseau routier sont données dans le Tableau 4.1, tandis que le Tableau 4.2 donne des informations sur les cinq IAVs.

TABLE 4.1 – Longueurs des routes

Route ($R_{i,j}$)	Longueur (mètre)	Route ($R_{i,j}$)	Longueurs (mètre)
$S_1 J_1$	27	$J_5 J_7$	32
$S_1 J_3$	70	$J_6 J_{10}$	44
$J_1 J_2$	39	$J_7 J_8$	32
$J_2 J_4$	111	$J_8 J_9$	26
$J_3 J_4$	45	$J_9 J_{10}$	29
$J_4 J_5$	20	$J_{10} J_{11}$	12
$J_5 J_6$	80	$J_{11} S_2$	40

TABLE 4.2 – Informations liées aux IAVs

IAV	État	Position relative		Vitesse (mètre/seconde)
		À	Distance (mètre)	
V_1	V_N	J_{10}	7	5
V_2	V_D	J_9	11	3
V_3	V_A	J_8	9	0
V_4	V_D	J_4	15	3,5
V_5	V_N	J_4	30	5

En appliquant la première méthode (le plus court parcours), nous constatons que le parcours $S_1; J_3; J_4; J_5; J_7; J_8; J_9; J_{10}; J_{11}; S_2$ est le chemin optimal (FIGURE 4.15), où la distance parcourue par le véhicule est de 306 mètres en un temps de 70 secondes.

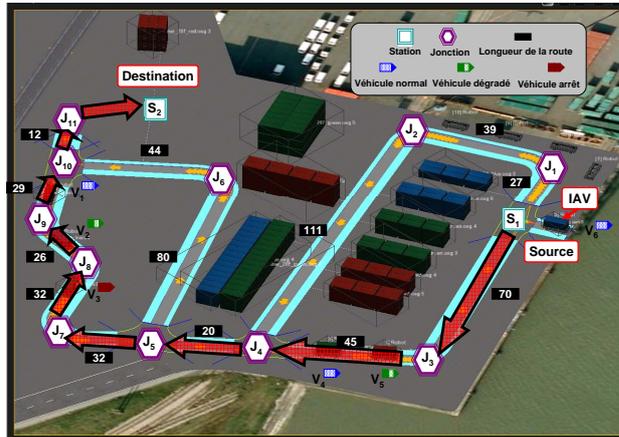


FIGURE 4.15 – Le plus court parcours

En pratique, le temps nécessaire pour parcourir ce chemin est ∞ parce que le véhicule V_3 est arrêté sur la route J_7J_8 , ce qui force le véhicule V_6 à s'arrêter après 41 secondes comme le montre la simulation présentée FIGURE 4.16.

En appliquant la deuxième méthode (moins de véhicules engagés), nous constatons que le parcours $S_1; J_1; J_2; J_4; J_5; J_6; J_{10}; J_{11}; S_2$ correspond au chemin optimal (FIGURE 4.17) en l'absence de véhicules. La distance parcourue est de 373 mètres en un temps de 84 secondes.

Ce parcours est meilleur que le premier ($84 \text{ secondes} < \infty$), malgré que la distance à parcourir soit plus longue ($373 \text{ mètres} > 306 \text{ mètres}$). La FIGURE 4.19 montre que le véhicule V_6 atteint la destination désirée après 84 secondes.

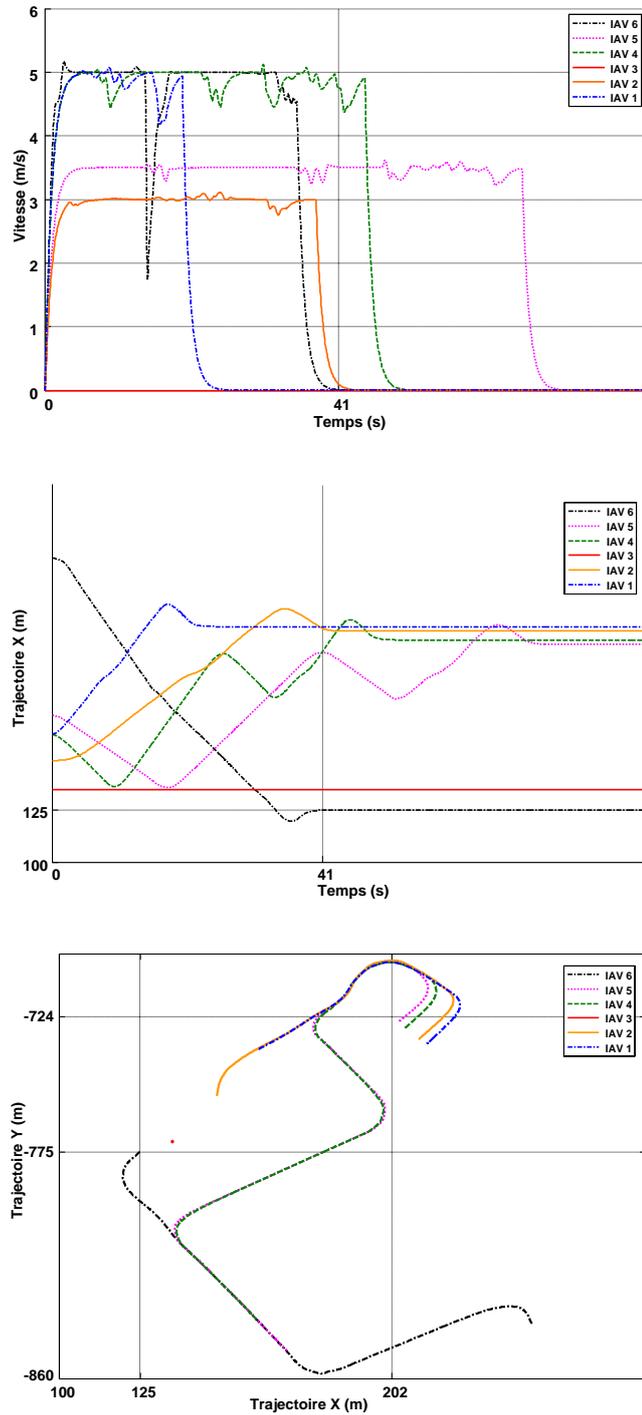


FIGURE 4.16 – Le véhicule V_6 ne peut pas atteindre sa destination

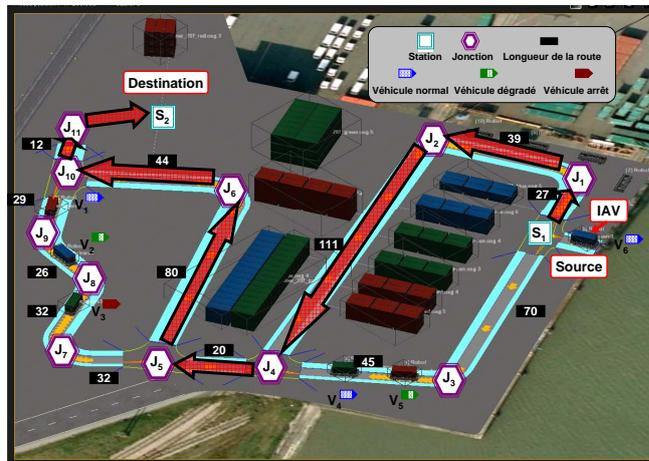


FIGURE 4.17 – Le chemin ayant le nombre minimal de véhicules

Après l'implémentation de l'algorithme d'optimisation proposé, nous trouvons que le parcours : $S_1; J_3; J_4; J_5; J_7; J_8; J_9; J_{10}; J_{11}; S_2$ correspond au chemin optimal (FIGURE 4.18), où la distance parcourue est de 311 mètres en un temps de 78 secondes < 84 secondes. Il correspond au meilleur résultat entre les trois possibilités.

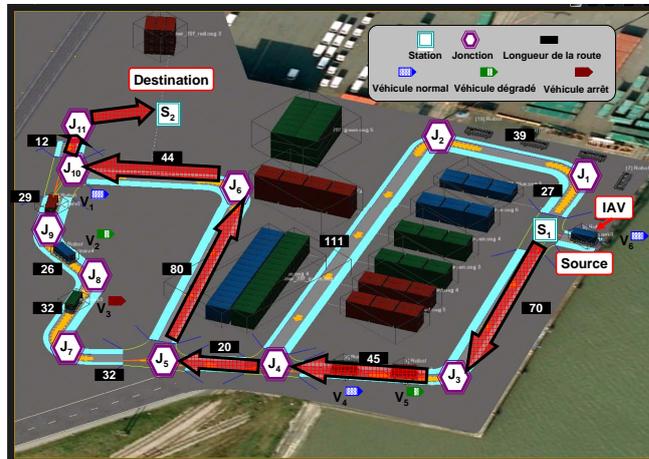


FIGURE 4.18 – Chemin optimal d'après l'algorithme proposé

La FIGURE 4.20 montre que le véhicule V_6 effectue le parcours en plus de 72 secondes, ce qui correspond au temps nécessaire pour atteindre la destination souhaitée dans des conditions normales. Le véhicule V_6 a dû adapter sa vitesse par rapport à l'état du véhicule V_5 , où la vitesse de V_6 est passée de 5 mètres/seconde à 3.5 mètres/seconde en un temps de $t=55$ secondes.

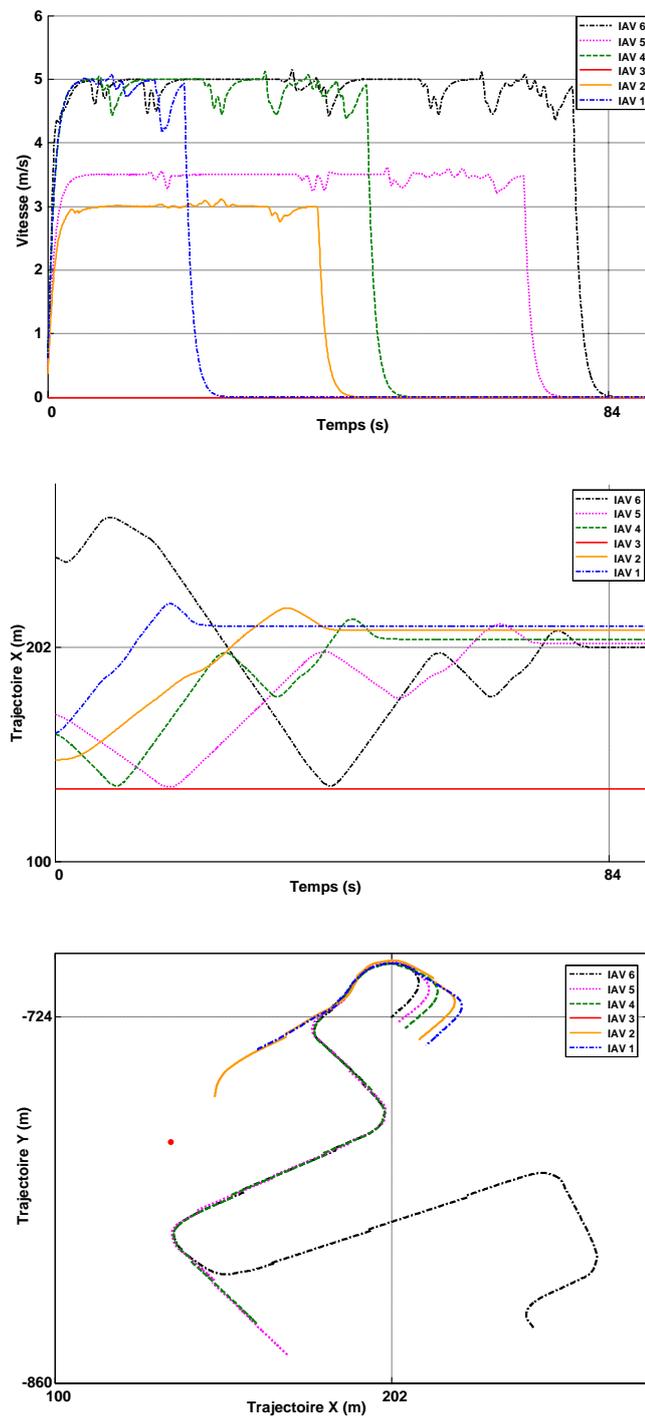


FIGURE 4.19 – Atteindre la destination S_2 après 84s

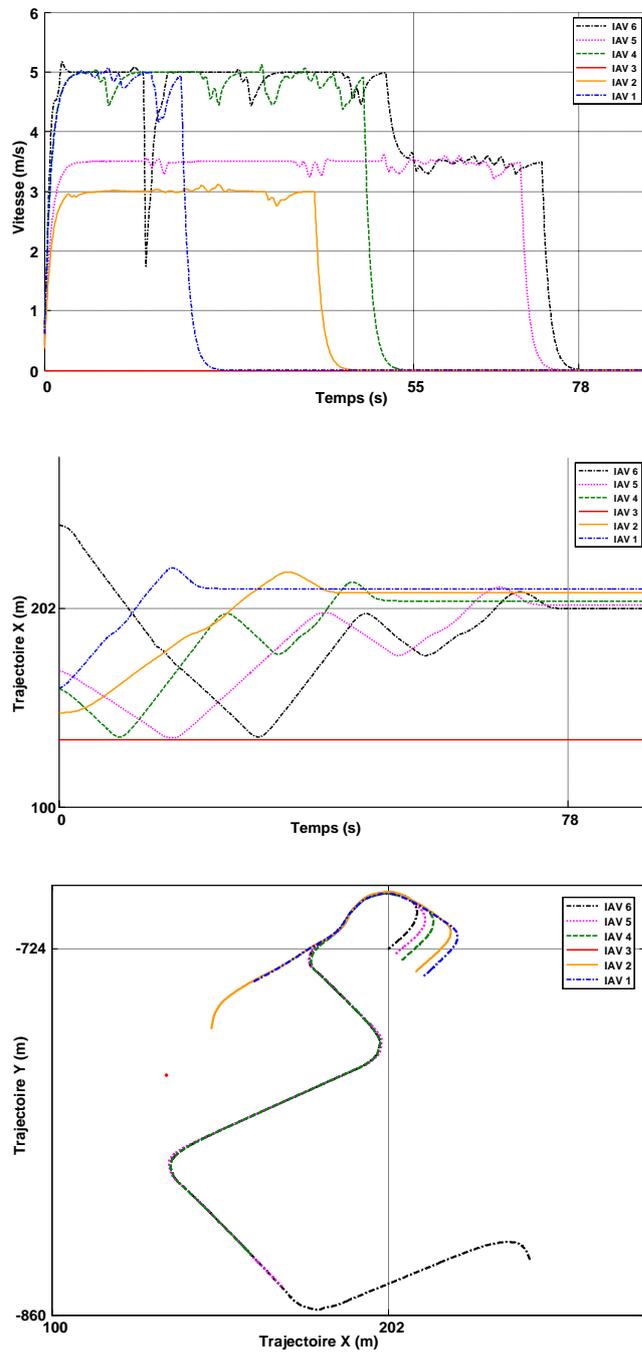


FIGURE 4.20 – Atteindre la destination S_2 en un temps optimal

4.7 Conclusion

Dans ce chapitre, un Modèle de Graphe Dynamique (MGD) d'un Système de Systèmes (SdS) est extrait à partir du modèle multi-niveaux à base de l'hypergraphe développé dans le chapitre 3. Ce modèle a permis d'élaborer une stratégie d'optimisation ciblée pour chaque composant système dynamique correspond à un niveau de modélisation donné. Cette stratégie basée sur les propriétés du MGD, aspect local (temps) et aspect global (évolution), permet sa décomposition en une succession de graphes statiques valués. L'optimisation en ligne d'un MGD est calculée à partir d'un algorithme qui reçoit en entrée les différents graphes statiques. Le MGD est illustré sur un système de transport intelligent (ITS) et l'algorithme d'optimisation est appliqué au système de routage de véhicules automatisés intelligents (IAVs). Une comparaison avec d'autres approches (FIGURE 4.21) a montré l'utilité de cette optimisation à base de MGD pour calculer, à chaque instant, le chemin à parcourir par un IAV, en fonction de l'état des différents composants systèmes de ITS.

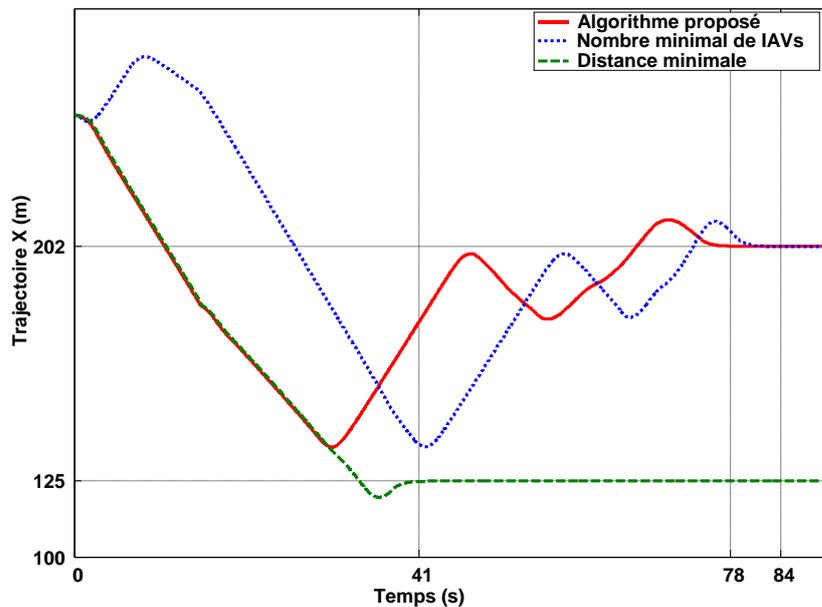


FIGURE 4.21 – Chemin optimal selon les trois méthodes étudiées

CHAPITRE 5

Conclusion et perspectives

Sommaire

5.1 Conclusion	113
5.2 Perspectives	114

5.1 Conclusion

Ce travail de recherche est réalisé dans le cadre d'une thématique nouvelle visant à développer des modèles graphiques pour la description de Systèmes de Systèmes (SdS). L'avantage d'une approche graphique réside dans sa généralité, où à partir du même modèle graphique (capable de représenter l'architecture du système), nous pouvons analyser les modes de fonctionnement des composants systèmes sur plusieurs niveaux, sans avoir à affronter de lourds calculs. Ceci a permis d'appliquer une stratégie d'optimisation multi-niveaux (verticale) pour la reconfiguration du SdS, et une optimisation à un seul niveau (transversale) pour la gestion optimale des fonctionnements indépendant, coopératif et évolutif des composants systèmes du même niveau.

Deux grandes contributions autour de la modélisation graphique de SdS découlent de ce travail de recherche.

1. La première contribution concerne le développement d'un modèle d'hypergraphe représentant sur plusieurs niveaux le concept de SdS. Ce modèle inspiré d'un hypergraphe imbriqué et valué permet de représenter les propriétés fondamentales d'un SdS que sont l'indépendance,

la dispersion, la coopération, et l'évolution. En exploitant les propriétés graphiques de l'hypergraphe, nous contribuons à la supervision. En parcourant le graphe du manière ascendante, c'est-à-dire, du niveau microscopique au niveau macroscopique, nous arrivons à suivre la propagation d'un changement de mode de fonctionnement d'un composant système sur l'ensemble du SdS. En parcourant le graphe dans l'autre sens, des solutions de reconfiguration peuvent être décelées. Le SdS peut alors continuer, en présence de défaillances, à assurer ses différentes missions, éventuellement avec des performances dégradées.

2. La deuxième contribution concerne le développement d'un modèle graphique dynamique, représentant un niveau de modélisation du SdS. Le but est d'élaborer une procédure d'optimisation en ligne, permettant de gérer en adaptatif les fonctionnements coopératif, indépendant et évolutif des composants systèmes associés à ce niveau de modélisation.

Ces modèles graphiques ont été validés en prenant comme support un système de transport intelligent (ITS) tel que proposé dans le cadre du projet InTraDE. Les composants systèmes de ce système ITS sont des véhicules intelligents autonomes (IAVs). Le modèle hypergraphe développé pour représenter le système ITS permet de suivre les conséquences de la défaillance d'un IAV sur la réalisation des missions globales. L'objectif d'un modèle de graphe dynamique est de gérer le trafic de composants systèmes IAVs au niveau macroscopique tout en respectant leurs propriétés d'indépendance, de coopération et d'évolution.

Les résultats de ces travaux de recherche ont été valorisés par l'implémentation d'algorithmes de supervision et d'optimisation en ligne sur un simulateur 3D en temps réel, pour une application de transport intelligent impliquant des véhicules IAVs opérant dans un environnement portuaire et confiné.

5.2 Perspectives

À la suite de notre travail de recherche initié en 2008, plusieurs perspectives futures de recherche dans le cadre du concept SdS sont apparues :

1. D'un point de vue modélisation, il est important de coupler le modèle de l'hypergraphe, dédié à la supervision de SdS à des modèles comportementaux multi-physiques, représentatifs des composants systèmes élémentaires au niveau microscopique. Un tel couplage va permettre d'associer à un instant donné le fonctionnement global d'un SdS au

fonctionnement d'un ou plusieurs éléments physiques d'un composant système élémentaire, c'est-à-dire, une intégration de modèle d'information (systèmes de communication) et de comportement (systèmes physiques).

2. Surveillance des deux types de systèmes :
 - i. les systèmes de communication entre véhicules (vehicle to vehicle : V2V) et entre véhicules et infrastructure (vehicle to infrastructure : V2I) : détection de défaut de liaisons de communication,
 - ii. les systèmes physiques : détection de défauts des composant formant le SdS.
3. La représentation de deux types de systèmes par un seul outil de modélisation : un modèle bond graph.

ANNEXES

ANNEXE **A**

Simulation 3D - temps réel

Sommaire

A.1 Pourquoi la simulation 3D - temps réel ?	119
A.2 Rôle du simulateur	120
A.3 Conception d'un scénario de simulation en temps réel	121
A.3.1 Mode terrain	121
A.3.2 Mode véhicule	122
A.3.3 Mode scénario	122
A.3.4 Mode simulation	122
A.3.5 Mode analyse	124
A.4 Implémentation de modules externes sur le simulateur	124

A.1 Pourquoi la simulation 3D - temps réel ?

La simulation 3D - temps réel, est une concaténation de modèles dynamiques avec un simulateur sous un format graphique en trois dimensions, et une communication temps réel avec un système physique. Cette combinaison de logiciel et matériel permet dans le cadre du projet InTraDE, de simuler le scénario de routage des véhicules intelligents autonomes (IAVs) dans un environnement virtuel et dynamique avant de transférer les consignes de trajectoires générées sur le véhicule réel. Dans un second temps, pendant la réalisation de sa mission, chaque véhicule IAV transmet au simulateur 3D - temps réel

les informations issues des capteurs embarqués afin que ce dernier arrive à reconstruire dans l'environnement virtuel le comportement réel du véhicule. Ainsi, cette plateforme servira comme moyen de supervision en ligne des systèmes IAVs.

Sur cette base, nous avons développé dans le cadre de ce travail de recherche, des algorithmes d'optimisation de trafic et de supervision sur une plateforme de simulation 3D - temps réel (FIGURE A.1) sur le logiciel SCANeR studio [OKTAL].



FIGURE A.1 – SCANeR studio

A.2 Rôle du simulateur

La plateforme SCANeR studio combine toutes les informations pertinentes en un seul système. La conception flexible et modulaire du système permet d'ajouter certaines extensions ou modules selon les besoins spécifiques. SCANeR studio est un logiciel de simulation de trafic routier qui permet :

- la création de terrains par superposition d'un modèle issu de bases de données cartographiques¹ et qui reprend exactement les paramètres réels et la disposition de différents objets sur une zone particulière,
- la modélisation de la dynamique de différents types de véhicules intervenant sur un terrain particulier,

1. par exemple Google Earth

- la gestion de trafic (contrôle des feux, priorité de circulation ...),
- l'implémentation d'algorithmes d'optimisation et de régulation de trafic à travers des scripts appropriés,
- la visualisation dynamique des objets 3D,
- la simulation des indicateurs sur le trafic et la dynamique des véhicules.

A.3 Conception d'un scénario de simulation en temps réel

A.3.1 Mode terrain

Le mode « Terrain » permet de créer un terrain en 3D suivant ses caractéristiques. En effet, il permet d'exploiter les informations issues d'une cartographie 3D d'un terrain afin de le reconstruire en virtuel. De plus, il offre une panoplie d'outils permettant de créer un profil particulier de route, de changer les dimensions des routes, le type de chaussée mais également les objets à placer comme les conteneurs dans les zones portuaires, les bâtiments, les ponts, les feux, les panneaux de signalisation Ces objets peuvent être positionnés comme dans un environnement réel (FIGURE A.2).

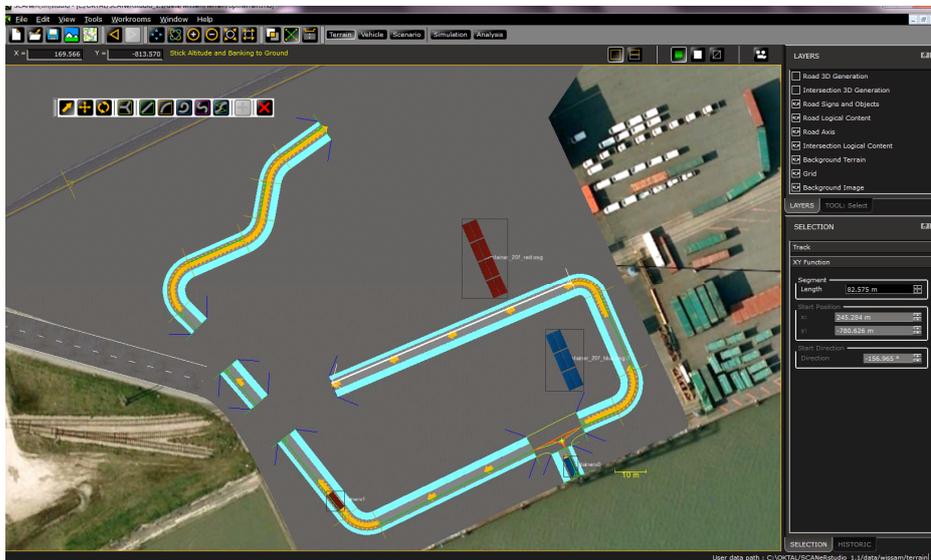


FIGURE A.2 – Mode terrain

A.3.2 Mode véhicule

Le mode « Vehicle » permet de créer un véhicule, de type léger ou lourd, avec toutes les dynamiques associées à la suspension, la pneumatique, la motorisation Ce mode permet également de définir la position et la vitesse initiales de chaque véhicule, son itinéraire, les capteurs embarqués, et le moyen de contrôle du véhicule à savoir véhicule autonome, interactif...(FIGURE A.3).

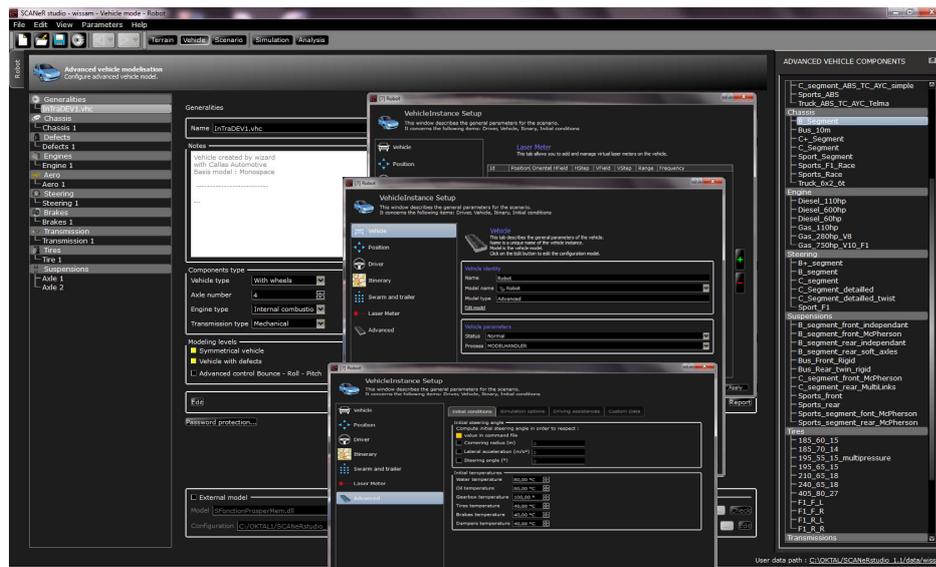


FIGURE A.3 – Mode véhicule

A.3.3 Mode scénario

Le mode « Scenario » est un mode qui permet de créer un scénario entre le terrain et les véhicules. En effet, on peut y ajouter des véhicules de différents types interactifs ou autonomes. De même, nous pouvons placer une source et un puits permettant de générer et de détruire un trafic aléatoire des véhicules. Ainsi nous pouvons créer des scénarios de trafic de véhicules avec différents niveaux de modélisation (FIGURE A.4).

A.3.4 Mode simulation

Le mode « Simulation » permet de lancer une simulation d'un scénario qu'on a spécifié au préalable. En effet, ce mode permet de superviser en 3D l'évaluation d'un ou de plusieurs véhicules avec la possibilité de tout visualiser en



FIGURE A.4 – Mode scénario

changeant le plan de vue. Pour cela, nous devons lancer un ou plusieurs modules de simulation, exemples, « Traffic » pour la gestion du trafic, « Physics » pour activer la dynamique de certains objets virtuels (conteneur), « Visual » pour la visualisation en 3D (FIGURE A.5).



FIGURE A.5 – Mode simulation

A.3.5 Mode analyse

Une fois la simulation effectuée, toutes les données relatives à chaque véhicule peuvent être affichées et analysées dans le mode « Analysis ». En effet, on peut récupérer les données issues des différents capteurs permettant ainsi d'avoir un rendu sur la dynamique d'un système de transport (FIGURE A.6).

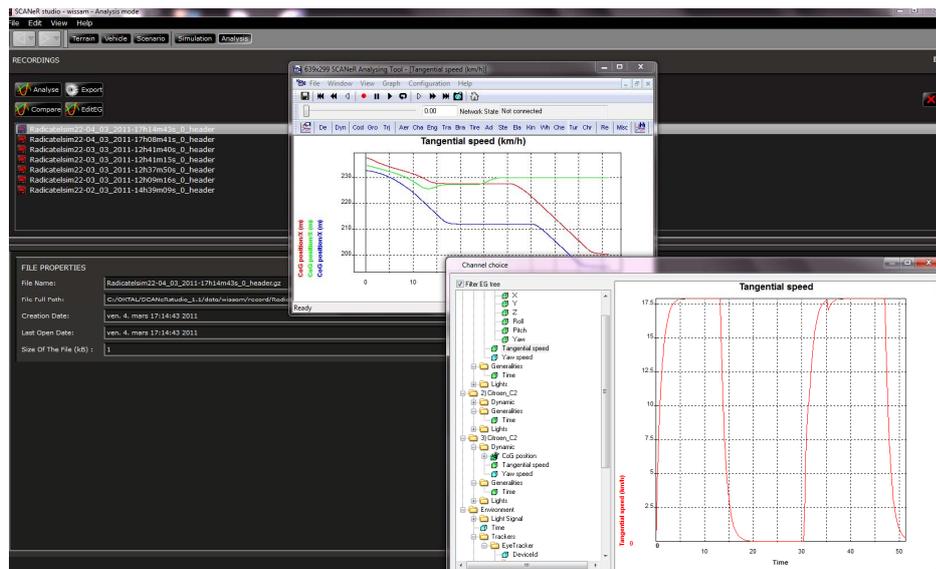


FIGURE A.6 – Mode analyse

A.4 Implémentation de modules externes sur le simulateur

La plateforme logiciel SCANeR studio de part sa flexibilité et sa compatibilité avec de nombreux langages de programmation (Java, C++, Python ...) offre la possibilité d'intégrer des modules externes, supplémentaires et dédiés à différentes tâches telle que la supervision à plusieurs niveaux (micro, méso, macro) du système de transport.

Ainsi l'environnement SCANeR studio se positionne d'une part comme un outil de co-simulation, permettant de faire cohabiter des simulateurs de technologies hétérogènes (Matlab, Simulink, FlexSim, MADKIT ...). Il est aussi utilisé comme une plateforme commune d'intégration de plusieurs algorithmes

de contrôle et commande, de diagnostic, d'optimisation, de supervision ... (FIGURE A.7).

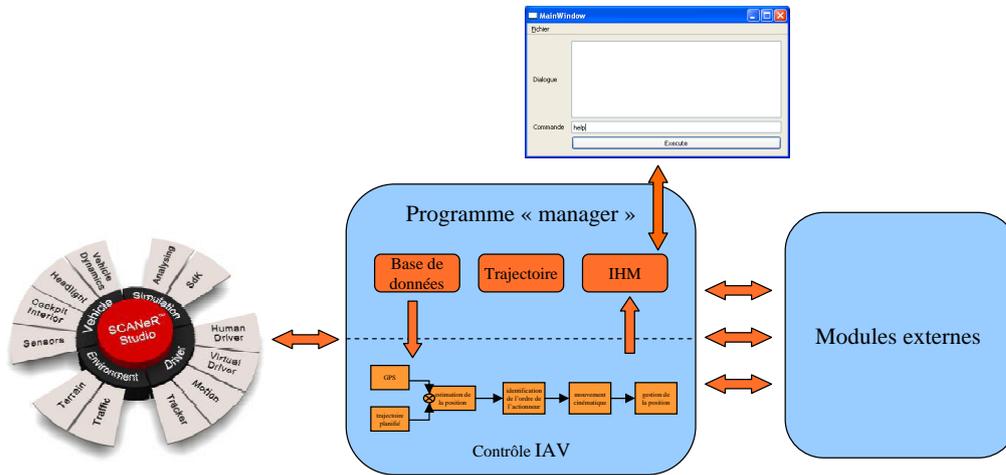


FIGURE A.7 – SCANer studio et les modules externes

ANNEXE B

Liste des publications scientifiques

B.1 Article de journal

- **Wissam KHALIL**, Rochdi MERZOUKI, Belkacem OULD-BOUAMAMA, & Hafid HAFFAF, *Hypergraph Models for System of Systems Supervision Design*, Transactions on Systems, Man, and Cybernetics, Part A, 2011, in press.

B.2 Articles de conférences internationales avec comité de lecture

- **Wissam KHALIL**, Rochdi MERZOUKI, Belkacem OULD-BOUAMAMA, & Hafid HAFFAF, *Model-Based Supervision of a Platoon of Intelligent and Autonomous Vehicles*, the 9th IEEE International Symposium on Safety, Security, and Rescue Robotics, November 1 - 5, 2011, KYOTO, JAPAN.
- **Wissam KHALIL**, Rochdi MERZOUKI, Belkacem OULD-BOUAMAMA, & Hafid HAFFAF, *System of Systems Architectural Modeling*, PAPYRUS Workshop on Fault Diagnosis and Fault Tolerant Control in large scale processing industries, October 6 - 7, 2011, CORSICA, FRANCE.
- **Wissam KHALIL**, Rochdi MERZOUKI & Belkacem OULD-BOUAMAMA, *Dynamic Model-Based for Intelligent Traffic Optimization inside Seaport Terminal*, the 14th International Conference on Harbor, Maritime & Multimodal Logistics Modelling and Simulation, pages 75 - 79, September 12 - 14, 2011, ROME, ITALY.

- Nizar CHATTI, **Wissam KHALIL**, Anne-Lise GEHIN, Rochdi MERZOUKI & Belkacem OULD-BOUAMAMA, *Intelligent Supervision of Autonomous Heavy Vehicles : Application to Maritime Area*, the 14th International Conference on Harbor, Maritime & Multimodal Logistics Modelling and Simulation, pages 33 - 37, September 12 - 14, 2011, ROME, ITALY.
- **Wissam KHALIL**, Rochdi MERZOUKI & Belkacem OULD-BOUAMAMA, *Modelling for Optimal Trajectory Planning of an Intelligent Transportation System*, the 7th IFAC Symposium on Intelligent Autonomous Vehicles, volume 7, part 1, September 6 - 8, 2010, LECCE, ITALY.
- **Wissam KHALIL**, Rochdi MERZOUKI & Belkacem OULD-BOUAMAMA, *Operating Management of Intelligent & Autonomous Vehcles*, IEEE Conference on Robotics, Automation and Mechatronics, pages 465 - 469, June 28 - 30, 2010, SINGAPORE.
- **Wissam KHALIL**, Rochdi MERZOUKI & Belkacem OULD-BOUAMAMA, *Dynamic Modeling of a Train of Intelligent Vehicles Inside a Confined Space*, the 12th IFAC Symposium on Control in Transportation Systems, pages 619 - 629, September 2 - 4, 2009, CALIFORNIA, USA.



BIBLIOGRAPHIE

- [ACKOFF, 1971] R. Ackoff, *Towards a system of systems concepts*, Management Science **17**(11), pages 661-672, 1971.
- [AHUJA, 1990] R.K. Ahuja, T.L. Magnanti, J.B. Orlin and R.E. Tarjan, *Faster algorithms for the shortest path problem*, Journal of ACM **37**, pages 213-223, 1990.
- [ALLIOT, 2002] J.M. Alliot, T. Schiex, P. Brisset et F. Garcia, *Intelligence artificielle et informatique théorique*, 2^{ème} édition, Cépaduès éditions, Toulouse, 2002.
- [ARMENGOL, 2000] J. Armengol, L.T. Massuy's, J. Vehi, J.L. de la Rosa, *A survey on interval model simulators and their properties related to fault detection*, Annual Reviews in Control **24**, pages 31-39, 2000.
- [ARMY, 2003] Army Acquisition Policy, Research, Development, and Acquisition. Army Regulation 70-1, U.S. Department of Defense, Department of the Army, Washington, DC, Decembre 31, 2003.
- [ARPINAR, 1999] I.B. Arpinar, U. Halici, S.Arpinar and A. Dogac, *Formalization of Workflows and Correctness Issues in the Presence of Concurrency*, Journal Distributed and Parallel Databases **7**(2), pages 199-248, 1999.
- [BAR-YAM, 2004] Y. Bar-Yam, M. A. Allison, R. Batdorf, H. Chen, H. Generazio, H. Singh and S. Tucker, *The characteristics and emerging behaviors system of systems*, NECSI : Complex Physical, Biological and Social Systems Project, <http://necsi.edu/education/oneweek/winter05/NECSISoS.pdf>, 2004.
- [BASSEVILLE, 1993] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes - Theory and Application*, Prentice-Hall, New Jersey, 1993.
- [BELLMAN, 1958] R. Bellman, *On a Routing Problem*, Quarterly of Applied Mathematics **16**(1), pages 87-90, 1958.

- [BEN-GAL, 2007] I. Ben-Gal, *Bayesian Networks*, Encyclopedia of Statistics in Quality and Reliability, edited by F. Ruggeri, R.S. Kennett, F.W. Faltin, Wiley, 2007.
- [BERGE, 1987] C. Berge, *Hypergraphes : Combinatoires des ensembles finis*, Gauthier-Villars, Paris, 1987.
- [BLANKE, 2006] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki and J. Schröder, *Diagnosis and Fault-Tolerant Control*, Springer-Verlag, New York, 2006.
- [BOCCALETTI, 2006] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez and D.U. Hwang, *Complex networks : Structure and dynamics*, Physics Reports **424**(4-5), pages 175-308, 2006.
- [BOARDMAN, 2006] J. Boardman and B. Sauser, *System of Systems - the meaning of of*, Proceedings of the 2006 IEEE/SMC, International Conference on System of Systems Engineering, pages 118-123, Los Angeles, April 2006.
- [BOULDING, 1956] K.E. Boulding, *General systems theory - The skeleton of science*, Management Science **2**(3), pages 197-208, 1956.
- [BUI-XUAN, 2003] B. Bui-Xuan, A. Ferreira, and A. Jarry, *Computing shortest, fastest, and foremost journeys in dynamic networks*, International Journal of Foundations of Computer Science **14**(2), pages 267-285, 2003.
- [CANTOT, 2009] P. Cantot and D. Luzeaux, *Simulation et modélisation des systèmes de systèmes : vers la maîtrise de la complexité*, Hermes - Lavoisier, Paris, 2009.
- [CARLOCK, 2001] P. G. Carlock and R. E. Fenton, *System of systems (SoS) enterprise systems engineering for information-intensive organizations*, Systems Engineering **4**(4), pages 242-261, 2001.
- [CARTIER, 2008] L. Cartier, *Le graphe comme outil pour enseigner la preuve et la modélisation*, Thèse de doctorat de l'Université Joseph Fourier - Grenoble I, 2008.
- [CHARARA, 2011] A. Charara, *Le Labex MS2T de l'UTC dans le peloton de tête de l'excellence scientifique*, l'Université de Technologie de Compiègne, <http://oasys.utc.fr/f-shic/spip/spip.php?article99>, 2011.
- [CHATTI, 2011-a] N. Chatti, A.L. Gehin, B. Ould-Bouamama and R. Merzouki, *Online Supervision of Intelligent Vehicle using functional and behavioural models*, IEEE Intelligent Vehicles Symposium (IV), pages 827-832, 2011.

- [CHATTI, 2011-b] N. Chatti, W. Khalil, A.L. Gehin, B. Ould-Bouamama and R. Merzouki, *Intelligent Supervision Of Autonomous Heavy Vehicles : Application To Maritime Area*, The 13th International Conference on Harbor, Maritime & Multimodal Logistics Modelling and Simulation, pages 33 - 37, 2011.
- [CHEMERO, 2008] A. Chemero and M.T. Turvey, *Autonomy and hypersets*, *BioSystems* **91**, pages 320-330, 2008.
- [CORMEN, 2001] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein. *Single-Source Shortest Paths*, Introduction to Algorithms, Second edition, Cambridge, pages 580-619, MIT Press, Massachusetts, 2001.
- [CORNUEJOLS, 2002] M. Cornuejols, *L'organisation des réseaux sémantiques verbaux et imagés : De l'analyse psycholinguistique au traitement automatique du langage*, *Bulletin de linguistique appliquée et générale* **27**, pages 39-56, 2002.
- [CORTES, 2003] C. Cortes, D. Pregibon and C. Volinsky. *Computational methods for dynamic graphs*, *Journal of Computational & Graphical Statistics* **12**(4), 950-970, 2003.
- [COURVEUR, 1988] J.M. Courveur and S. Haddad, *Towards a general and powerful computation of flows for parameterized coloured nets*, 9th European Workshop on Application and Theory of Petri Nets **II**, Venice, Italy, June 1988.
- [COWELL, 1999] R.G. Cowell, A.P. Dawid, S.L. Lauritzen and D.J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer-Verlag, Berlin, 1999.
- [DAHMAN, 2008] J. Dahmann, *Systems Engineering for Department of Defense Systems of Systems*, System of Systems - Innovation for the 21st Century, 9th chapter, edited by M. Jamshidi, publisher John Wiley & Sons Inc, New York, 2008.
- [DARPA] *Defense Advanced Research Projects Agency (DARPA) Urban Challenge*, <http://www.darpa.mil/index.html>.
- [DAVID, 1997] R. David et H. Alla, *Du Grafset aux réseaux de Pétri*, 2^{ème} édition, Hermès-Lavoisier, Paris, 1997.
- [DECHTER, 2007] R. Dechter and R. Mateescu, *And/or search spaces for graphical models*, *Artificial Intelligence* **171**(2-3), pages 73-106, 2007.
- [DECHTER, 2003] R. Dechter, *Constraint Processing*, Morgan Kaufmann Series in Artificial Intelligence, Elsevier, Burlington, 2003.
- [DECKER, 1999] S. Decker, M. Erdmann, D. Fensel and R. Studer, *Ontobroker : ontology based access to distributed and semi structured information*, edited by R. Meersman et al., Semantic Issues in Multimedia

- Systems (DS-8), Kluwer Academic Publisher, Boston, pages 351-369, 1999.
- [DELAURENTIS, 2006] D. DeLaurentis and R.K. Callaway, *A System-of-Systems Perspective for Future Public Policy*, Review of Policy Research **21**(6), pages 829-837, 2006.
- [DELAURENTIS, 2005] D. DeLaurentis, *Understanding Transportation as a System-of-Systems Design Problem*, 43rd AIAA Aerospace Sciences Meeting, Reno, Nevada, January 2005.
- [DEMETRESCU, 2006] C. Demetrescu and G.F. Italiano, *Fully dynamic all pairs shortest paths with real edge weights*, Journal of Computer and System Sciences **72**(5), pages 813-837, 2006.
- [DEO, 2007] N. Deo and A. Cami, *Preferential deletion in dynamic models of web-like networks*, Information Processing Letters **102**(4), pages 156-162, 2007.
- [DIAZ, 2001] M. Diaz, *Les réseaux de Pétri, modèles fondamentaux*, Hermès, Science Publications, Paris, 2001.
- [DIJKSTRA, 1959] E.W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische Mathematik **1**, pages 269-271, 1959.
- [DION, 2003] J.M. Dion, C. Commault and J.V.D. Woude, *Generic properties and control of linear structured systems : a survey*, Automatica **39**, pages 1125-1144, 2003.
- [DJEZIRI, 2009] M.A. Djeziri, R. Merzouki and B. Ould-Bouamama, *Robust Monitoring of an Electric Vehicle with Structured and Unstructured Uncertainties*, IEEE Transactions on Vehicular Technology **58**(9), pages 4710-4719, 2009.
- [DUFFY, 2008] M. Duffy, B. Garrett, C. Riley and D. Sandor, *Future Transportation Fuel System of Systems*, System of Systems - Innovation for the 21st Century, 17th chapter, edited by M. Jamshidi, publisher John Wiley & Sons Inc, New York, 2008.
- [EISNER, 1991] H. Eisner, J. Marciniak and R. McMillan, *Computer-aided system of systems (C2) engineering*, IEEE Conference on Systems, Man, and Cybernetics, Charlottesville, VA, 1991.
- [EULER, 1736] L. Euler, *Solutio problematis ad geometriam situs pertinentis*, Commentarii academiae scientiarum Petropolitanae **8**, pages 128-140, 1736.
- [FERREIRA, 2002] A. Ferreira, *On models and algorithms for dynamic communication networks : The case for evolving graphs*, 4^{ème} rencontres

- francophones sur les Aspects Algorithmiques des Telecommunications (ALGOTEL'2002), Mèze, France, 2002.
- [FRIESE, 2006] P. Friese and J. Rambaub, *Online-optimization of multi-elevator transport systems with reoptimization algorithms based on set-partitioning models*, Discrete Applied Mathematics **154**, pages 1908-1931, 2006.
- [GAINES, 1995] B.R. Gaines, M.L.G. Shaw, *Collaboration through concept maps*, Proceedings of CSCL95, Computer Supported Cooperative Learning, Bloomington, 1995.
- [GAINES, 1994] B.R. Gaines, M.L.G. Shaw, *Concept maps indexing multimedia knowledge bases*, AAAI94 Workshop : Indexing and reuse in multimedia systems, Menlo Park, pages 36-45, California, 1994.
- [GALLO, 1986] G. Gallo and S. Pallottino. *Shortest path methods : A unifying approach*, Mathematical Programming Study **26**, pages 38-64, 1986.
- [GEHIN, 2008] A.L. Gehin and M. Staroswiecki, *Reconfiguration analysis using generic component models*, IEEE Transactions on Systems Man and Cybernetic-Part A : Systems and Humans **38**(3), pages 575-583, 2008.
- [GENRICH, 1981] H.J. Genrich and K. Lautenbach, *System modeling with high-level Petri-nets*, Theoretical Computer Science **13**, pages 109-136, 1981.
- [GLOBAL, 2007] Global Security Organization, <http://www.globalsecurity.org/military/systems/ground/fcs-back.htm>, 2007.
- [GROHE, 2006] M. Grohe, *The structure of tractable constraint satisfaction problems*, Proceedings of the 31st Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science **4162**, pages 58-72, 2006.
- [HADDAD, 1987] S. Haddad, *Une catégorie régulière de réseau de Pétri de haut niveau : définition, propriétés et réductions. Application à la validation de systèmes distribués*, thèse de doctorat de l' Université Pierre et Marie Curie, Paris, 1987.
- [HIPEL, 2008] K. Hipel, A. Obeidi, L. Fang and D.M. Kilgour, *Sustainable Environmental Management From A System Of Systems Engineering Perspective* System of Systems Engineering, 11th chapter, Edited by Mo Jamshidi, publisher John Wiley & Sons Inc, New York, 2008.
- [HOLLAND, 1995] J.H. Holland, *Hidden Order : How Adaptation Builds Complexity*, Reading, Addison-Wesley, New York, 1995.

- [InTraDE] Website of InTraDE - <http://www.intrade-nwe.eu>.
- [JACOB, 1974] F. Jacob, *The Logic of Living Systems*, Allen Lane, London, 1974.
- [JACKSON, 1984] M.C. Jackson and P. Keys, *Towards a system of systems methodologies*, The Journal of the Operational Research Society **35**(6), pages 473-486, 1984.
- [JAMSHIDI, 2009] M. Jamshidi, *Systems of Systems Engineering : Principles and Applications*, CRC Press, Boca Raton, 2009.
- [JAMSHIDI, 2008] M. Jamshidi, *System of Systems Engineering : Innovations for the 21st Century*, Wiley Series in Systems Engineering and Management, publisher John Wiley & Sons Inc, New York, 2008.
- [JCIDS, 2001] JCIDS, *Joint capabilities integration and development instructions*, 2001.
- [JENSEN, 1991] K. Jensen, *Coloured Petri nets : A high level language for system design and analysis*, High-level Petri Nets, Theory and Application, Edited by Jensen and Rozenberg, pages 44-119. Springer-Verlag, Berlin, 1991.
- [KHEIRBEK, 1995] A. Kheirbek and Y. Chiaramella, *Integrating hypermedia and information retrieval with conceptual graphs*, Hypertext-Information Retrieval-Multimedia, edited by R. Kuhlen and M. Rittberger, Proceedings HIM'95, Konstanz, pages 47-60, Konstanz, Germany, 1995.
- [KORDIK, 2008] V. Kordik, *Petri net, theory and applications*, I-Tech Education and Publishing, Vienna, 2008.
- [KRYGIEL, 1999] A.J. Krygiel, *Behind the Wizard's Curtain : An Integrated Environment for a System of Systems*, National Defense University and the DoD C4ISR Cooperative Research Program, July 1999.
- [LE, 2011] Q.A. Le, G. Strylewicz and J.N. Doctor, *Detecting Blood Laboratory Errors Using a Bayesian Network : An Evaluation on Liver Enzyme Tests*, Medical Decision Making **31**(2), pages 325-337, 2011.
- [LE-ANH, 2006] T. Le-Anh and M.B.M. De Koster, *A review of design and control of automated guided vehicle systems*, European Journal of Operational Research **171**(1), pages 1-23, 2006.
- [LJUNG, 1999] L. Ljung, *System Identification - Theory For the User*, 2nd edition, Prentice Hall, Upper Saddle River, New Jersey, 1999.
- [LUSKASIK, 1998] S.J. Luskasik, *System, systems of systems, and the education of engineers*, Artificial Intelligence for Engineering Design, Analysis and Manufacturing **12**(1), pages 55-60, 1998.

- [LUZEAUX, 2008-a] D. Luzeaux et J.R. Ruault, *Systèmes de systèmes : concepts et illustrations pratiques*, Hermès-Lavoisier, Paris 2008.
- [LUZEAUX, 2008-b] D. Luzeaux et J.R. Ruault, *Ingénierie des systèmes de systèmes-méthodes et outils*, IC2 informatique et systèmes, Hermes Science Publications, Paris, 2008.
- [LYNCE, 2002] I. Lynce, L. Baptista and J. Marques-Silva, *Complete unrestricted backtracking algorithms for satisfiability*, Proceedings of the International Symposium on Theory and Applications of Satisfiability Testing, pages 214-221, Cincinnati, Ohio, USA, 2002.
- [MAHR, 2010] T. Mahr, J. Srour, M. De Weerd and R. Zuidwijk, *Can agents measure up? A comparative study of an agent-based and on-line optimization approach for a drayage problem with uncertainty*, Transportation Research Part C : Emerging Technologies **18**(1), Pages 99-119, 2010.
- [MAIER, 1996] M. Maier, *Architecting principles of systems-of-systems*, Proceedings of the Sixth Annual International Symposium INCOSE, pages 567-574, INCOSE, Boston, 1996.
- [MANTHORPE, 1996] W.H. Manthorpe, *The emerging joint system of system : A systems engineering challenge and opportunity for APL*, John Hopkins APL Technical Digest **17**(3), pages 305-310, 1996.
- [MARSHALL, 2002] A.H. Marshall, S.I. McClean, C.M. Shapcott and P.H. Millard, *Modelling Patient Duration of Stay to Facilitate Resource Management of Geriatric Hospitals*, Health Care Management Science **5**(4), 313-319, 2002.
- [MARTIN, 1999] Ph. Martin and P. Eklund, *Embedding Knowledge in web documents : CGs versus XML-based metada languages*, Proceedings of ICCS 1999, 7th International Conference on Conceptual Structures, Springer Verlag, LNAI 1640, pages 230-246, Blacksburg, 1999.
- [MARTIN, 1996] Ph. Martin, *Exploitation de graphes conceptuels et de documents structurés et hypertextes pour l'acquisition de connaissances et la recherche d'informations*, thèse de doctorat de l'Université de Nice - Sophia Antipolis, 1996.
- [MAURYA, 2004] M. R. Maurya, R. Rengaswamy and V. Venkatasubramanian, *Application of signed digraphs-based analysis for fault diagnosis of chemical process flowsheets*, Engineering Applications of Artificial Intelligence, **17**(5), pages 501-518, 2004.
- [MEDJAHER, 2008] K. Medjaher, A. Mechraoui and N. Zerhouni, *Diagnostic et pronostic de défaillances par réseaux bayésiens*, Manuscrit auteur,

- 4èmes Journées Francophones sur les Réseaux Bayésiens, JFRB'2008 , Lyon, 2008.
- [MERTENS, 2006] K. Mertens, *An ant-based approach for solving dynamic constraint optimization problems*, PhD thesis, Katholieke Universiteit Leuven, Leuven, 2006.
- [MERZOUKI, 2009] R. Merzouki, M.A. Djeziri and B. Ould-Bouamama, *Intelligent Monitoring of Electric Vehicle*, IEEE/ASME Conference on Advanced Intelligent Mechatronics, AIM2009, pages 797-804, Singapore, July 2009.
- [MERZOUKI, 2007] R. Merzouki, K. Medjaher, M.A. Djeziri and B. Ould-Bouamama, *Backlash Fault Detection in Mechatronics System*, Mechatronics Journal **17**(6), pages 299-310, 2007.
- [MUKHERJEE, 2004] A. Mukherjee, *Dynamic Stochastic Optimization Models for Air Traffic Flow Management*, Ph.D. thesis, University of California, Berkeley, 2004.
- [MULLER, 2000] P.A. Muller and N. Gaertner, *Modélisation objet avec UML*, Eyrolles, Paris, 2000.
- [MURATAT, 1989] T. Muratat, *Petri Nets : properties, analysis and application*, Proceedings of the IEEE **77**(4), pages 541-580, 1989.
- [NANAYAKKARA, 2010] T. Nanayakkara, F. Sahin and M. Jamshidi, *Intelligent Control Systems with an Introduction to System of Systems Engineering*, CRC Press, Boca Raton, 2010.
- [OKTAL] Website of OKTAL - <http://www.oktal.fr>.
- [OWENS, 1995] A.W.A. Owens, *The emerging U.S. system of systems*, US Naval Institute Proceedings, and "introduction" in Dominant Battlespace Knowledge, edited by S. Johnson and M. Libicki, National Defense University Press, Washington 1995.
- [PALLOTTINO, 1997] S. Pallottino and M. G. Scutella', *Shortest path algorithms in transportation models : classical and innovative aspects*, Technical Report TR-97-06, Università di Pisa - Dipartimento di Informatica, 1997.
- [PALMIERI, 2010] F. Palmieri, U. Fiore and S. Ricciardi, *A GRASP-based network re-optimization strategy for improving RWA in multi-constrained optical transport infrastructures*, Computer Communications **33**(15), pages 1809-1822, 2010.
- [PEI, 2000] R. Pei, *System of systems integration (SoSI)-A smart way of acquiring army C4I2WS systems*, Proceedings of the 2000 Summer Computer Simulation Conference, pages 574-579, 2000.

- [PETRI, 1962] C.A. Petri, *Kommunikation mit Automaten*, Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962.
- [QUILLIAN, 1968] M.R. Quillian, *Semantic Memory*, In *Semantic Information Processing*, edited by M. Minski, MIT Press, Cambridge, 1968.
- [RAMCHANDANI, 1974] C. Ramchandani, *Analysis of asynchronous concurrent systems by timed Petri nets*, Technical report **120**, Laboratory for computer science, Project MAC, TR 120, MIT, Cambridge, 1974.
- [RAMALINGAM, 1996] G. Ramalingam and T.W. Reps, *An incremental algorithm for a generalization of the shortest-path problem*, *Journal of Algorithms* **21**(2), pages 267-305, 1996.
- [REISIG, 1991] W. Reisig, *Petri nets and algebraic specifications*, *Theoretical Computer Science* **80**, pages 1-34, 1991.
- [REVIEW] Review of Maritime Transport, United Nation Conference On Trade and Development, Geneva, 2009. Available at http://www.unctad.org/en/docs/rmt2009_en.pdf.
- [ROTSHEIN, 2005] A.P. Rotshtein, M. Posner and H.B. Rakytyanska, *Cause and effect analysis by fuzzy relational equations and a genetic algorithm*, *Reliability Engineering and System Safety* **91**(9), pages 1095-1101, 2005.
- [SAGE, 2001] A.P. Sage and C.D. Cuppan, *On the systems engineering and management of systems of systems and federations of systems*, *Inf., Knowl., Systems Management* **2**(4), pages 325-345, 2001.
- [SAMANTARAY, 2008] A.K. Samantaray and B. Ould-Bouamama, *Model-based Process Supervision, A Bond Graph Approach*, Series : *Advances in Industrial Control*, Springer, Berlin 2008.
- [SANCHO, 2001] G. Sancho, *Adaptation architecturale de logiciels collaboratifs dans les environnements ubiquitaires. Contribution à l'interopérabilité par la sémantique*, thèse de doctorat de l'université de Toulouse, 2010.
- [SHENHAR, 1994] A. Shenhar, *A new systems engineering taxonomy*, *Proceedings of the 4th International symposium of the National Council on System Engineering* **2**, pages 261-276, 1994.
- [SHIBASAKI, 2008] R. Shibasaki and J. Pearlman, *Global Earth Observation System of Systems*, *System of Systems Engineering*, Edited by Mo Jamsidi, publisher John Wiley & Sons Inc, New York, 2008.
- [SRIDHAR, 2007] P. Sridhar, A.M. Madni and M. Jamsidi, *Hierarchical aggregation and intelligent monitoring and control in fault-tolerant wireless sensor networks*, *IEEE Systems Journal* **1**(1), pages 38-54, 2007.

- [STAROSWIECKI, 2001] M. Staroswiecki and G. Comtet-Varga, *Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems*, Automatica **37**(5), pages 687-699, 2001.
- [STEFANOVITCH, 2010] N. Stefanovitch, *Contributions à la résolution de problèmes d'optimisation de contraintes distribuées dynamiques à l'aide de modèles graphiques pour la coordination multiagents*, thèse de doctorat de l' Université Pierre et Marie Curie, Paris, 2010.
- [SUEUR, 1989] C. Sueur and G. Dauphin-Tanguy, *Structural controllability / observability of linear systems represented by bond-graphs*, Journal of Franklin Institute, **326**(6), pages 869-883, 1989.
- [TIEN, 2008] J.M. Tien, *A System of Systems View of Services*, System of Systems Engineering, 13th chapter, Edited by Mo Jamshidi, publisher John Wiley & Sons Inc, New York, 2008.
- [TOSCANO, 2005] R. Toscano, *Commande et Diagnostic des Systèmes Dynamiques*, ellipses, Paris, 2005.
- [VALLET, 2005] D. Vallet, M. Fernandez and P. Castells, *An ontology-based information retrieval model*, Lecture Notes in Computer Science **3532**, edited by A. Gómez-Pérez and J. Euzenat, pages 455-470, Springer, Berlin, 2005.
- [VERFAILLIE, 2005] G. Verfaillie and N. Jussien, *Constraint solving in uncertain and dynamic environments : A survey*, Constraints **10**(3), pages 253-281, 2005.
- [VLIET, 1978] D.V. Vliet, *Improved shortest path algorithms for transport networks*, Transportation Research **12**(1), pages 7-20, 1978.
- [WALKER, 2007] D. Walker, *Realizing a corporate SOSE Environment*, Key-note presentation, 2007 IEEE SoSE Conference, San Antonio, TX, USA, 2007.
- [WICKRAMASINGHE, 2008] N. Wickramasinghe, S. Chalasani, R.V. Boppana and A.M. Madni *Healthcare System of Systems*, System of Systems Engineering, Edited by Mo Jamshidi, publisher John Wiley & Sons Inc, New York, 2008.

Contribution à la modélisation graphique de système de systèmes

Le travail de recherche initié dans cette thèse de doctorat s'inscrit dans le contexte du projet européen InTraDE (Intelligent Transportation for Dynamic Environment). Les aspects théoriques développés concernent la modélisation graphique de Système de Systèmes (SdS). Ces derniers peuvent être modélisés sur plusieurs niveaux, allant du niveau microscopique au niveau macroscopique. Une approche à base d'hypergraphe est utilisée pour représenter les différents niveaux hiérarchiques d'un SdS ainsi que les propriétés fondamentales qui le définissent : l'indépendance, la coopération, la dispersion, et l'évolution des composants systèmes qui le constituent. La modélisation graphique proposée permet d'élaborer une stratégie de supervision en ligne en suivant les parcours ascendant et descendant de l'hypergraphe.

Pour gérer le fonctionnement global des composants systèmes à un niveau de modélisation donné, un modèle graphique dynamique est extrait de l'hypergraphe. Ce dernier, permet d'appliquer une stratégie d'optimisation en ligne. Appliquée au système de transport intelligent, cette approche permet de déterminer à chaque instant le temps que mettra le véhicule autonome et intelligent pour atteindre sa cible lorsqu'il évolue dans un environnement évoluant en permanence.

Mots-clés : Système de Systèmes, Modélisation, Hypergraphe, Graphe Dynamique, Systèmes de Transport Intelligents.

Contribution to graphical modelling of system of systems

This work has been realized in the framework of the European project InTraDE (Intelligent Transportation for Dynamic Environment). The main scientific contribution focuses on graphical modeling of System of Systems (SoS) in the engineering field. To model such systems, a graphical method based on hypergraph approach is proposed. Hypergraphs are used to describe the multiple hierarchical levels of a SoS and to describe its associated fundamental properties such as : the independence, cooperation, dispersion and the evolution of the component systems. Through this graphical modeling, online supervision strategies can be directly deduced.

To manage the global operating of the component systems at a specific level, a dynamic graphical model is extracted from the hypergraph. This later allows the elaboration of an online optimization strategy. As a case study, an intelligent transportation system is considered. An online optimization algorithm currently updates, the time used for each intelligent autonomous vehicle to reach its target, in a permanently evolving environment.

Keywords : System of Systems, Modelling, Hypergraph, Dynamic Graph, Intelligent Transportation Systems.