

Intelligence Artificielle pour l'aide à la décision des systèmes dynamiques : Diagnostic, Prévision, Recommandation d'actions

Christine Largouët

▶ **To cite this version:**

Christine Largouët. Intelligence Artificielle pour l'aide à la décision des systèmes dynamiques : Diagnostic, Prévision, Recommandation d'actions. Intelligence artificielle [cs.AI]. Université de Rennes 1, 2019. tel-02437159

HAL Id: tel-02437159
https://inria.hal.science/tel-02437159

Submitted on 13 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HABILITATION À DIRIGER DES RECHERCHES

Université de Rennes 1
Mention : Informatique

École doctorale : MATH-STIC

présentée par

Christine Largouët

Intelligence Artificielle pour l'aide à la décision des systèmes dynamiques :

Diagnostic, Prévision, Recommandation d'actions

Date de soutenance : le 2 décembre 2019

Composition du jury :

Frédérick Garcia	DR INRA	MIAT INRA Toulouse
Florence Le Ber	ICPEF HDR	ENGESS Strasbourg (rapporteure)
Alexandre Termier	Professeur	Université de Rennes 1, IRISA
Maguelonne Teisseire	DR IRSTEA	UMR TETIS Montpellier (rapporteure)
Sergio Yovine	DR CNRS	Universidad ORT Uruguay (rapporteur)
Pascale Zaraté	Professeure	Université Toulouse 1 Capitole, IRIT

Remerciements

La légende veut que les pages de remerciements soient les plus lues des mémoires de thèse ou de HDR. En espérant ne pas décevoir tous ceux qui m'ont accompagnée durant ces années...

Tout d'abord je tiens à remercier les membres du jury sans qui cette HDR n'aurait pu avoir lieu. Mes rapporteurs Florence le Ber, Maguelonne Teisseire et Sergio Yovine qui tous trois ont accepté promptement de relire ce manuscrit ainsi que Frédérick Garcia et Pascale Zaraté qui, tout comme Florence Le Ber, m'ont fait le plaisir de venir à Rennes dans cette période de déplacements compliqués en ce mois de décembre 2019.

Mes premiers remerciement de chercheuse se tournent vers Marie-Odile Cordier qui m'a accompagnée durant de longues années. J'ai eu la chance de travailler auprès d'une grande dame de l'Intelligence Artificielle. J'ai beaucoup appris à ses cotés tout en admirant son intuition, sa culture du domaine, ses raisonnements et sa capacité de conviction dans les projets pluridisciplinaires où il s'agissait de promouvoir nos approches souvent considérées comme (trop) originales.

Alexandre Termier, qui lui a succédé à la tête de l'équipe Dream devenue Lacodam, a réussi à me faire passer tout en douceur vers une thématique plus orientée fouille de données dans laquelle je m'y retrouve. Merci à lui pour cette gestion bienveillante de l'équipe, ses encouragements, sa motivation communicative, et ses conseils me permettant de terminer plus sereinement cette HDR. Je suis ravie qu'il ait accepté d'être le président de ce jury.

Gérard Douaire est le premier à m'avoir donné l'opportunité de faire de la recherche en me proposant une thèse à Agrocampus qui s'est poursuivie par un poste de maître de conférences. Je le remercie pour son soutien et de sa présence, avec Madeleine Douaire, le jour de la soutenance alors que de nombreuses autres activités les sollicitent aujourd'hui.

Puisqu'il n'est pas toujours facile d'être une maître de conférences en informatique dans un école d'agronomie, je tiens à remercier Agrocampus (Grégoire Thomas), l'IRISA (Jean-Marc Jézéquel, François Bodin et Bruno Arnaldi) et INRIA (Stéphane Ubéda) pour m'avoir toujours soutenue ma démarche de rattachement à l'IRISA me permettant d'effectuer ce travail de recherche dans d'excellentes conditions.

Un grand merci également à Guy Fontenelle, mon gestionnaire des pêches préféré, rencontré fortuitement à l'IRD de Nouméa alors que nous étions tous deux des enseignants-chercheurs d'Agrocampus en délégation qui ne se connaissaient pas. C'est certainement grâce à cette grande ouverture d'esprit qui le caractérise que nous avons pu travailler et communiquer aussi facilement alors que nous venions de deux mondes différents. Yves-Marie Bozec, de retour du Mexique et en partance pour l'Australie a également fait une escale à Agrocampus pour jouer le rôle, pas toujours facile, de l'écologue plongé dans un laboratoire d'informaticiens. J'espère qu'il aura pris autant de plaisir que moi à notre travail commun et je garde de très bons souvenirs de notre périple en Tasmanie pour présenter les tout premiers débuts d'Ecomata.

Les étudiants que j'ai encadrés sont partie prenante de ce travail, aussi je remercie chaleureusement Alban Grastien, Yulong Zhao, Alban Siffer et Raphaël Gauthier qui furent des doctorants aussi compétents que sympathiques. C'est une des chances de notre métier de pouvoir travailler avec de jeunes étudiants que l'on voit évoluer sur trois années et qui nous apportent certainement autant que nous leur apportons.

Un grand merci à Louis Bonneau de Beaufort, mon compagnon du quotidien à Agrocampus qui sait rendre, par la variété de ses sujets de discussions, les journées toujours intéressantes malgré notre isolement. J'apprécie de pouvoir compter sur lui pour toutes les activités d'enseignement, son investissement, la justesse de ses remarques et ses nombreuses idées sont l'occasion de nombreux échanges indispensables à notre travail.

C'est toujours un plaisir de retrouver toute l'équipe Lacodam et la bonne humeur permanente qui règne dans le groupe aussi bien dans les réunions de travail (qui savent malgré tout rester efficaces), que dans les pauses café-gâteaux, les déjeuners-points d'équipe ou les séminaires Tahiti. Merci à mes collègues de longue date : Laurence Rozé et Véronique Masson que je viens souvent déranger pour parler de boulot et de bien d'autres choses encore, à ma nouvelle collègue de bureau, Elisa Fromont, pour son énergie incroyable et contagieuse, très avisée et de bon

conseil et, qui me pardonne toujours de l'avoir enfermée à l'extérieur du bureau. Merci à tous les autres membres de l'équipe, les permanents, Tassadit Bouadi, Philippe Besnard, Luis Galarraga et tous les doctorants pour cet état d'esprit si chaleureux que vous savez influer au sein de l'équipe. Enfin, je tiens à remercier les deux assistantes qui m'apportent leur soutien avec efficacité et gentillesse tout au long de l'année, Héléna Rouillé d'Agrocampus et Gaëlle Tworkowski d'INRIA.

Pour finir, je ne saurais oublier mes deux garçons, Solal et Ulysse, pour leur prévenance et leur compréhension face à mon manque de disponibilité pendant cette période de HDR, mes oublis récurrents de l'heure qui tourne conduisant à des diners bien trop tardifs pour des écoliers. Enfin, je remercie Stéph pour son soutien et ses encouragements, sa présence le jour de la soutenance malgré un emploi du temps "over-booké", et son éternelle capacité à me faire relativiser.

Table des matières

Introduction **7**

1 Aide à la décision et Intelligence Artificielle **15**
1.1 Aide à la décision : origines et théories 16
1.2 Quelques définitions 19
1.3 L'Intelligence Artificielle pour l'aide à la décision 24

2 Diagnostic de systèmes à événements discrets **29**
2.1 Définitions et travaux de référence 30
2.1.1 Des modèles pour expliquer 30
2.1.2 Terminologie 31
2.1.3 Systèmes à événements discrets 33
2.1.4 Définitions formelles et originales du diagnostic de SED . . . 35
2.1.5 Approches décentralisée et distribuée 39
2.2 Diagnostic incrémental et décentralisé 40
2.2.1 Représentation 41
2.2.2 Diagnostic incrémental 43
2.2.3 Diagnostic décentralisé 44
2.2.4 Diagnostic de systèmes reconfigurables 45
2.3 Autres méthodes de diagnostic 46
2.3.1 Diagnostic par model-checking 46
2.3.2 Diagnostic avec ASP 55

3 Model-checking pour la gestion des écosystèmes **59**
3.1 Enjeu 60
3.2 Modélisation qualitative des écosystèmes 60
3.3 Problématique 62
3.4 Framework de modélisation 63
3.4.1 Modélisation par automates temporisés 63
3.4.2 Génération automatique du modèle d'automates 66
3.4.3 L'écosystème marin du lagon d'Ouvéa 69

3.5 Requêtes prédictives *what-if ?* par model-checking 72
3.5.1 Patrons de requêtes 72
3.5.2 Logiciel EcoMata 75
3.5.3 Expérimentations 76
3.5.4 Bilan sur l'exploration de modèles par des scénarios prédictifs 77
3.6 Requêtes proactives *what-for ?* par synthèse de contrôleur 79
3.6.1 Automate temporisé de jeu et à coûts 79
3.6.2 Synthèse de contrôleur 81
3.6.3 Planification de stratégies 83
3.7 Intégration de facteurs socio-économiques 85
3.8 Conclusion 88

4 Fouille de données et Aide à la décision 89
4.1 Contexte Scientifique 89
4.2 Détection des anomalies à partir d'un flux de données 90
4.3 Machine-Learning en Agro-écologie 94
4.3.1 Tendances actuelles pour la gestion des agro-écosystèmes . . 95
4.3.2 Travaux en cours : prédiction de la ration alimentaire des truies allaitantes 99
4.4 Comprendre ses données pour mieux les traiter 102
4.4.1 Test d'unimodalité 103
4.4.2 Recherche du k dans la méthode des k plus proches voisins . 105

5 Bilan et Perspectives 107
5.1 Bilan 107
5.2 Perspectives 111
5.2.1 Apprentissage de modèles de comportement interprétables . 111
5.2.2 Modélisation des facteurs socio-économiques et de la résilience 113

Bibliographie 131

Curriculum Vitæ détaillé 133

Introduction

La rédaction d'un mémoire d'habilitation n'est pas nécessairement l'activité la plus naturelle pour le chercheur ou l'enseignant-chercheur, plus habitué à se projeter vers l'avenir. L'idée consiste à prendre un peu de recul par rapport à son quotidien, à s'isoler dans la bulle narcissique de son propre parcours, afin de revenir sur des années de recherche depuis la fin du doctorat, de poser une analyse sur les travaux passés, de tenter d'harmoniser le tout, même si la recherche aujourd'hui a évolué et que le chemin suit parfois des opportunités, qu'elles soient humaines ou financières. Si l'exercice peut paraître un peu dissuasif, par manque de temps évidemment, d'envie de se replonger dans des choses anciennes, de conviction d'un travail suffisant, il suffit cependant de se lancer pour se laisser prendre au jeu. C'est l'occasion de réexaminer d'un regard prospectif et critique les travaux menés et de réévaluer les perspectives au regard de l'expérience et de l'évolution de la recherche.

Ma thématique de recherche se place dans le domaine de l'Intelligence Artificielle, et ce bien avant que la folie médiatique ne s'empare du sujet, du temps même où le sujet paraissait un peu d'arrière-garde. Je n'ai pourtant jamais regretté cette orientation qui fait la part belle au raisonnement, s'intéresse aux hommes et à leur manière de penser, avant de se préoccuper de la technique. Je me suis intéressée à la modélisation de systèmes dynamiques complexes avec pour objectif le diagnostic, la supervision et l'aide à la recommandation d'actions. L'ambition consiste à fournir des méthodes, des algorithmes et des logiciels permettant de réaliser ces tâches sur des systèmes difficiles à représenter par des approches de l'informatique classique. Maître de conférences en informatique à Agrocampus Ouest, j'ai orienté une grande partie de mes travaux vers des approches de modélisation des écosystèmes dans le but de "mieux comprendre pour mieux gérer". Cette préoccupation avait pour idée première de pouvoir rester proche des thématiques de mes collègues enseignant-chercheurs en agro-écologie et de mes étudiants, élèves ingénieurs ; elle s'est transformée, avec l'expérience, en appétence tant les applications environnementales me semblent offrir un support illustratif unique pour l'informatique et en particulier pour l'Intelligence Artificielle, aussi bien dans la caractérisation de ces problèmes complexes, que dans les réelles attentes de solutions innovantes. Si une

bonne partie des travaux présentés dans ce mémoire a donc été suggérée directement par des applications, il m'a toujours paru essentiel, en tant que chercheuse en informatique, de proposer des méthodes suffisamment génériques pour répondre à une classe de problèmes dans le domaine agro-écologique mais pas seulement.

Contexte

Le contexte est celui des systèmes complexes, qu'ils soient naturels ou techniques, pour lesquels la dimension temporelle est prépondérante aussi bien pour leur représentation que pour leur exploration. Cette information temporelle, nécessaire à la définition de la dynamique du système, peut néanmoins être variable ou mal connue. Les systèmes sont généralement décentralisés dans le sens où ils sont constitués d'un ensemble de sous-systèmes interagissants (agents, composants, espèces, etc.) pouvant éventuellement coopérer dans le but de répondre à un même but. Selon les travaux abordés durant cette période, les systèmes considérés ont pu être déterministes, dans le cadre des activités portant sur le diagnostic par exemple, ou non-déterministes lorsque je me suis intéressée à la gestion des écosystèmes. Ces derniers peuvent être des systèmes mal ou partiellement connus, pour lesquels les données issues de capteurs sont coûteuses ou impossible à obtenir (c'est en particulier le cas des écosystèmes marins). L'expertise humaine doit alors pouvoir être intégrée au modèle. En ce qui concerne les interactions avec l'extérieur, le système peut entretenir différents types d'échanges. Des observations peuvent être recueillies par des capteurs qui informent de l'état du système. Des actions peuvent également être effectuées sur le système : des actions contrôlables comme par exemple des commandes de l'utilisateur ou des perturbations anthropiques qui peuvent être appliquées sur un écosystème, ou encore des actions non contrôlables telles que les pannes dans des systèmes industriels ou les catastrophes naturelles qui touchent les écosystèmes.

Pluridisciplinarité

La particularité de mon travail touche à la pluridisciplinarité, souvent encouragée mais compliquée à pratiquer. La pluridisciplinarité apparait lorsqu'il s'agit d'examiner un système d'une autre discipline décrit selon un certain langage, qui peut être mathématique ou oral de la part des experts, afin d'envisager, dans un premier temps, les potentialités d'une collaboration. Mes contributions se sont concentrées pour une grande part dans le domaine de l'agro-écologie, l'halieutique en particulier avec la gestion des pêches ou l'agriculture pour des problèmes relatifs au pâturage des bovins ou à l'alimentation des truies allaitantes. Certaines tentatives de collaboration pluridisciplinaire sont cependant parfois vaines : le système peut être trop complexe (en particulier sur les aspects spatio-temporels des systèmes écologiques) ou ne comporte pas de caractéristiques suffisamment intéressantes, et

des techniques déjà éprouvées pourraient résoudre le problème moyennant un peu d'ingéniérie. La première précaution d'une collaboration pluridisciplinaire consiste donc à vérifier que chacune des parties soit satisfaite : que le chercheur en informatique ait matière à étudier de nouvelles méthodes, que le chercheur thématicien participe à une approche innovante qui lui propose des meilleurs résultats que ses outils actuels. La grande satisfaction reste cependant l'ouverture intellectuelle avec la découverte, parfois longue, d'autres disciplines et des problématiques insoupçonnées, ainsi que des modes de pensées variés.

Questions scientifiques

Je me suis intéressée aux systèmes dynamiques que l'on peut considérer comme difficiles à appréhender soit par leur taille, la nature hétérogène des données disponibles ou les processus de différentes natures qui les composent, avec pour objectif d'aider au processus de décision comme la supervision ou la conception de stratégie de gestion. Quels que soient les sujets abordés dans ce mémoire, ma préoccupation constante aura été de les traiter selon le point de vue de l'Intelligence Artificielle en proposant des modèles intelligibles dotés d'un fort pouvoir explicatif et autorisant l'interaction avec l'utilisateur ou le gestionnaire.

1- Modélisation qualitative des systèmes dynamiques

Mon premier objectif aura tout d'abord été de modéliser ces systèmes en tenant compte de leur particularité temporelle et en utilisant les données disponibles sur le système, que ce soit des connaissances expertes avec toute l'imprécision qui en découle, ou des modèles numériques connus dont il s'agit alors d'extraire des informations pertinentes. Forts de l'expérience des systèmes à événements discrets (SED) pour le diagnostic à base de modèles, la première question qui se pose est de savoir si une modélisation qualitative à l'aide de ce formalisme s'avère adaptée lorsqu'on y ajoute des contraintes temporelles. Comment intégrer, dans un formalisme suffisamment générique, tous les sous-systèmes qui peuvent être variés par leur type de dynamiques ou par les informations dont on dispose sur eux ? Il est reconnu que la difficulté des approches à base de SED réside dans la construction des modèles qui peuvent devenir rapidement trop volumineux pour être construits à la main. Les problématiques concernent donc la construction de modèles et la réduction de la complexité inhérente.

2- Méthodes d'exploration des modèles

Mon deuxième objectif a consisté à développer des méthodes s'appuyant sur cette représentation qualitative afin de réaliser plusieurs tâches. Je me suis intéressée à l'exploitation du modèle selon plusieurs points de vue : le diagnostic, la supervision, la prédiction, la recommandation d'actions et enfin la planification. Pour une majeure partie de ces problèmes, la question s'est posée de l'usage de méthodes

formelles telles que le model-checking et la synthèse de contrôleur pour résoudre efficacement ces problèmes. Le model-checking et la synthèse de contrôleur sont des techniques bien connues pour leur efficacité dans le domaine de la vérification de systèmes temps réels. La problématique à laquelle je me suis intéressée est la reformulation de ces problèmes d'IA à l'aide des méthodes symboliques. En ce qui concerne le domaine théorique du diagnostic, après les avancées du diagnostic des systèmes décentralisés des années 2000, la question s'est posée de pouvoir diagnostiquer, en ligne, des systèmes reconfigurables qui sont des systèmes dont la topologie peut évoluer au fil du temps.

3- Aide à la décision
Développer des méthodes d'exploration dans le but d'aider l'utilisateur ou le gestionnaire à prendre la meilleure décision est le troisième objectif. La problématique consiste non seulement à répondre au besoin exprimé mais également à proposer les éléments de connaissances pertinents permettant l'interprétation des résultats. Cette interprétation est nécessairement contrainte par les choix d'abstraction et de formalisation du modèle mais également par les événements et les alternatives d'action. Il s'agit de proposer des éléments de compréhension du fonctionnement du système réel en s'appuyant sur des représentations compréhensibles, comme la notion de trajectoire pour le diagnostic par exemple. Pour les problèmes de gestion des écosystèmes, le problème consiste à intégrer plus activement l'utilisateur dans la boucle en lui offrant la possibilité d'exprimer des hypothèses jusqu'à évaluer la solution la plus satisfaisante (qui n'est pas nécessairement l'optimale). Mais la recherche de stratégies optimales selon des contraintes posées par l'utilisateur reste également une problématique intéressante et complémentaire des précédentes approches, surtout pour les systèmes difficiles à analyser par la simple prédiction.

4- Conception d'outils
Enfin un dernier objectif qui me paraissait nécessaire, puisque l'informatique consiste malgré tout à développer des logiciels, est d'implémenter ces techniques de représentation et d'exploitation de modèles sur des applications réelles. Il s'agit de proposer des outils au service des décideurs. Malgré le choix de solliciter des techniques reconnues pour leur efficacité, le passage à l'échelle sur des applications réelles pose question. Les outils proposés satisferont-ils leurs utilisateurs sur des cas concrets ?

Présentation du mémoire

J'ai choisi d'organiser ce document selon le schéma classique de présentation de la recherche en informatique en l'articulant selon les thèmes des sous-disciplines de

l'Intelligence Artificielle auxquels ils se rapportent. Les objectifs précédents sont traités en totalité ou plus partiellement selon les chapitres.

Pour commencer ce mémoire, il me semblait important de resituer le contexte de l'aide à la décision qui est le fil directeur de ce mémoire. Le premier chapitre est consacré à ce sujet en présentant, après les diverses théories, les définitions de référence sur le sujet. Les différents domaines de l'Intelligence Artificielle et leur manière de traiter le problème de l'aide à la décision sont abordés en fin de chapitre.

Le second chapitre s'intéresse au problème du diagnostic de systèmes dynamiques à l'aide des systèmes à événements discrets. Après une présentation du domaine, je décris les contributions réalisées durant mes premières années de recherche en particulier l'approche décentralisée et incrémentale du diagnostic. Le diagnostic à base de modèles s'appuie sur une représentation qualitative à l'aide d'automates et le problème d'aide à la décision consiste à déterminer les pannes d'un système en proposant une trajectoire décrivant le comportement du système. Une deuxième partie du chapitre est consacrée aux travaux que j'ai menés sur le diagnostic à l'aide de méthodes formelles, en particulier les travaux à l'aide du model-checking ayant posé des bases de réflexion pour la suite des travaux qui élargissent le cadre du diagnostic.

Le troisième chapitre aborde une démarche plus ascendante qui, partant des problèmes de gestion des agro-écosystèmes, propose des méthodes pouvant se généraliser à une classe plus large de systèmes. Une modélisation qualitative des écosystèmes à l'aide d'automates temporisés est définie ainsi que différentes approches pour aider le processus d'aide à la décision : des scénarios *what-if ?*, pour répondre aux problèmes de prédiction, aux scénarios *what-for ?* pour la recommandation d'actions. L'exploration des modèles repose sur le model-checking et la synthèse de contrôleur des automates temporisés. Ces travaux ont conduit au développement de deux logiciels, EcoMata et PaturMata, pour la gestion des écosystèmes marins et la recherche de stratégies de pâturages.

Enfin le quatrième chapitre aborde un tournant dans mes recherches et présente des travaux plus récents puisqu'il s'agit d'intégrer les méthodes de fouilles de données dans les travaux d'aide à la décision. Une première partie est consacrée au domaine de l'agro-écologie et le second à des méthodes de détection des anomalies ou de l'unimodalité dans les données.

Le cinquième chapitre conclut ce mémoire en présentant un bilan des contributions scientifiques et présente le projet de recherche que je souhaite poursuivre dans la continuité des idées défendues dans ce mémoire. En annexe, un curriculum détaillé ainsi qu'une bibliographie personnelle sont donnés.

Parcours

Pour conclure cette introduction, je présenterai mon parcours à la lumière duquel il sera sans doute plus aisé de comprendre les travaux menés durant ces années de recherche qui suivent le doctorat.

J'ai été recrutée en 2001 à Agrocampus Ouest (Rennes) en tant que maître de conférences en informatique après une thèse sous le statut d'AERC (Assistante d'Enseignement et de Recherche Contractuelle) dans ce même établissement. Agrocampus Ouest regroupe deux formations d'ingénieurs en agronomie (en 3 ans après les classes préparatoires BCPST) et en agroalimentaire (en 5 ans) ainsi que de nombreux masters. Bien que maître de conférences à Agrocampus Ouest j'ai toujours été rattachée à l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) pour mes activités de recherche, que ce soit durant ma thèse ou après mon recrutement. Jeune recrutée, j'ai poursuivi mes activités de recherche à l'IRISA au sein de l'équipe AIDA dirigée par Jacques Nicolas. L'équipe développait alors une activité très forte autour du diagnostic à base de modèles (surtout sur des applications de télécommunications). Profitant du dynamisme autour de cette thématique, je me suis intéressée au diagnostic à l'aide de méthodes formelles et au problème des systèmes reconfigurables.

En 2003, un poste de maître de conférences en informatique en délégation à l'Université de la Nouvelle-Calédonie a retenu mon attention puisqu'il s'agissait de rejoindre, pour la partie recherche, l'équipe US Espace de l'IRD de Nouméa qui initiait des recherches en informatique pour gérer les écosystèmes. Par ailleurs, après six ans d'expérience dans une école d'ingénieurs agronomes, j'aspirais à enseigner, pour un temps, à l'université. Durant deux années et demi, j'ai participé aux activités de recherche de cette équipe naissante de l'IRD dont l'objectif consistait à représenter des phénomènes complexes afin de proposer aux décideurs une assistance intelligente pour une gestion raisonnée de l'environnement. Dans ce cadre, j'ai travaillé sur des projets variés mais cependant relatifs à l'environnement comme la modélisation dynamique des risques de pollution pour l'aide à la gestion de l'eau douce ou encore le suivi de trajectoires des récifs coralliens. Durant cette période j'ai également continué quelques travaux sur le diagnostic lors de visites respectives avec des anciens collègues de l'IRISA en poste à l'ANU de Canberra (Sylvie Thiébaux et Yannick Pencolé). Une collaboration de longue durée, près de dix ans, a été initiée avec Guy Fontenelle, professeur en halieutique, en poste également à l'IRD de Nouméa avec qui j'ai entrepris des recherches sur la modélisation des

écosystèmes marins.

De retour à Rennes début 2006 à l'issue de ma délégation, j'ai réintégré mon poste à Agrocampus Ouest. Jusqu'en 2011, le laboratoire informatique d'Agrocampus Ouest de Rennes n'avait pas de statut officiel concernant ses activités de recherche et je travaillais alors à l'IRISA en tant que "collaborateur extérieur". En 2011, j'ai proposé au directeur de l'IRISA, François Bodin, un projet de recherche permettant d'intégrer les membres du laboratoire à cette UMR, nous offrant une structure plus confortable pour mener nos recherches (collègues informaticiens, infrastructures techniques, manifestations scientifiques), et défendu cette position auprès de la direction d'Agrocampus Ouest. Les membres de l'unité pédagogique informatique d'Agrocampus ont donc été rattachés à l'IRISA qui est devenue notre UMR officielle, Agrocampus Ouest devenant établissement tutelle de notre équipe. Nous avons été, depuis cette date, membres de l'équipe DREAM dirigée par Marie-Odile Cordier jusqu'en 2014, puis après son départ à la retraite par Alexandre Termier, professeur à l'université de Rennes 1. En janvier 2016, l'équipe DREAM devient l'équipe LACODAM et oriente ses recherches sur l'extraction automatique de connaissances à partir de gros volumes de données. Durant cette période j'ai poursuivis mes travaux sur la modélisation des écosystèmes puis ai récemment orienté ma thématique vers la fouille de données pour l'aide à la décision.

Chapitre 1

Aide à la décision et Intelligence Artificielle

Sommaire

1.1 Aide à la décision : origines et théories 16
1.2 Quelques définitions 19
1.3 L'Intelligence Artificielle pour l'aide à la décision . . . 24

Dans son roman, *Le Joueur d'échecs*, Stefan Zweig exprime tout à la fois la curiosité et l'incompréhension éprouvées pour les joueurs d'échecs dont l'esprit est entièrement dévoué à la réflexion du meilleur coup : *"Comment se figurer l'activité d'un cerveau exclusivement occupé, sa vie durant, d'une surface composée de soixante-quatre cases noires et blanches ?"*. Les échecs ont toujours été un laboratoire parfait pour étudier la prise de décision, on parle même de "psychologie échiquéenne" : Comment le joueur prend-il une décision ? Combien de coups a-t-il d'avance ? Comment anticiper les réactions de son adversaire ? Comment savoir s'il a détecté un piège ? Malgré la présence des ordinateurs entrés en lice depuis des années, pour l'entrainement ou l'analyse, les échecs restent passionnants car le joueur est encore seul face à son adversaire. La solitude du joueur d'échecs devient pourtant de moins en moins courante dans notre vie de tous les jours. Alors qu'autrefois l'ami pouvait être le meilleur conseiller, nous cherchons aujourd'hui à nous entourer des meilleurs experts pour prendre nos décisions quotidiennes. Le fleurissement des *coachs* pour nous aider à faire les meilleurs choix sur des sujets de la vie de tous les jours nous révèle à quel point prendre une décision peut-être difficile et important.

Nous nous intéressons ici au problème de l'aide à la décision comme problématique scientifique. Cette thématique apparait dans les situations complexes et dotées d'un fort enjeu pour lesquelles le "bon sens" ou les calculs élémentaires ne sont plus suffisants pour aider à décider. Des systèmes informatiques, qui s'appuient sur une approche plus formelle de cette situation, peuvent être utilisés pour éclairer ce choix. Bien sûr cette représentation a pour inconvénient de proposer une "abstraction" qui lisse parfois tous les aspects ambigus de la réalité et l'enjeu consiste donc à trouver la formalisation la plus adaptée et la manière de l'utiliser. Partant de cette représentation, nous pouvons alors nous poser la question de savoir à qui est-ce que l'on cherche à apporter de l'aide ? à l'homme ou à la machine ? Les problématiques traitées dans ce mémoire, s'intéressent à proposer des solutions pour aider l'homme à prendre ses propres décisions et s'inscrivent donc uniquement dans la thématique, certes vaste, mais bien définie qu'est *l'aide à la décision*, et non dans celle des *systèmes intelligents* où les machines prennent les décisions de manière autonome. Puisque *"les hommes réfléchissent avant d'agir"* [Pom97], l'Intelligence Artificielle et ses aptitudes à reproduire le raisonnement humain, peut être naturellement partie prenante de ce processus de décision. Les solutions proposées dans ce manuscrit seront donc traitées du point de vue de l'IA reliant le raisonnement à la décision.

Ce chapitre introductif vise à présenter le domaine de l'aide à la décision, à donner les définitions qui nous semblent les plus appropriées aujourd'hui, avant de s'attarder un peu plus spécifiquement sur les approches de type Intelligence Artificielle.

1.1 Aide à la décision : origines et théories

Même si des travaux sur les probabilités pour la décision ont été menés au XVIII^{e} siècle, on attribue généralement les débuts de l'aide à la décision à l'armée britannique avec l'apparition du terme "Operation Research", traduit par "recherche opérationnelle" [Bow03], durant la seconde guerre mondiale. De nombreux scientifiques anglais et américains ont été appelés pour faire de la recherche sur des opérations militaires comme par exemple le développement de méthodes pour la conduite des convois, l'organisation des opérations anti-sous-marins ou la recherche des emplacements favorables aux radars pour la meilleure détection des appareils ennemis.

Entre les années 40 et 50, le domaine de la recherche opérationnelle n'est plus restreint au monde militaire et s'étend au monde civil avec des utilisations variées telles que l'approvisionnement, la logistique, les constructions. Deux avancées ma-

jeures, l'algorithme du simplexe pour la programmation linéaire développé par George Dantzig [Dan90], et le développement des ordinateurs, permettent de généraliser la résolution de problème aux systèmes de grande taille. À cette époque pourtant, le terme d'aide à la décision n'était pas encore employé. Il faut attendre 1947 et la thèse de Herbert Simon [Qui16], économiste et sociologue, dans laquelle il constate que les comportements des décideurs au sein des organisations sont difficiles à mettre en œuvre avec les méthodes classiques d'aide à la décision, regroupées dans ce que l'on appelle la *théorie rationnelle*. Dans sa théorie de la *rationalité limitée*, Simon exprime que la plupart des décisions humaines effectivement appliquées correspondent à des choix satisfaisants et que rarement les choix optimaux, souvent irréalistes ou inutilisables, sont mis en pratique [Kal10]. Sans le savoir encore à cette époque, Simon abordait des principes fondamentaux qui rejoindraient plus tard ceux de l'Intelligence Artificielle.

Les années 50-60 et la "taylorisation" du travail correspondent à l'âge d'or de la recherche opérationnelle et des techniques d'aide à la décision. Durant ces mêmes années, l'aide à la décision va s'enrichir par les travaux sur la théorie des ensembles flous de Zadeh [Zad65], permettant d'accroître l'expressivité des langages formels, et par les réflexions autour des relations de préférences (cf. la synthèse [Tso08] pour plus de détails). De son coté, Simon s'est associé à Newell afin de proposer deux programmes de ce que l'on appelle enfin l'Intelligence Artificielle, "the Logic Theorist" [NS72] et "the General Problem Solver" [NS63], qui permettent de résoudre des problèmes par la représentation des connaissances et le raisonnement. Ces nouveaux travaux commencent à s'écarter de l'approche classique de la recherche opérationnelle où il s'agit toujours de maximiser la valeur d'une fonction sur les conséquences des actions potentielles. Les années 60 voient également apparaître les premiers travaux sur la prise de décision à partir de plusieurs critères, éventuellement conflictuels entre eux. Le français Bernard Roy, qui a de nombreuses expériences dans le milieu industriel, est un précurseur du domaine [DD11]. Von Neumann et Morgenstern [VNM47] ainsi que Savage [Sav54] proposent un modèle pour les décisions économiques dans lesquelles l'incertitude et le risque doivent être considérés. L'intérêt de ce modèle est de représenter l'incertitude sur l'état du système à l'aide de probabilités sur des événements incertains, et de classer les conséquences des actions par l'espérance d'utilité. Le modèle d'espérance d'utilité subjective de Savage a connu un fort succès puisqu'il permet de ramener toute situation d'incertitude à une situation de risque, l'incertitude étant probabilisée.

La discipline connait des difficultés à la fin des années 70. Dans un célèbre article, Ackoff annonce le déclin de la recherche opérationnelle, "The future of the Operational Research is past" [Ack79]. Les modèles trop théoriques, trop ob-

jectifs ne sont pas appliqués car trop éloignés de la réalité [Bou03]. Une nouvelle méthodologie s'appuyant sur les sciences cognitives rompt complètement avec les approches classiques basées sur l'optimisation de fonctions. La différence majeure réside sur une prise en considération plus importante de la formulation du problème d'aide à la décision, considérée comme prépondérante dans les approches cognitives. Une attention particulière est portée aux décideurs afin de définir le problème de décision pour toutes les personnes concernées et d'envisager les impacts sur leur comportement [WWF74, Roy93]. Ces approches ont eu pour conséquence de se focaliser non plus sur la décision elle-même mais sur le processus de décision [BDPP09]. Les systèmes d'aide à la décision (ou "Decision Support Systems" en anglais) commencent à apparaître, dans des années 80, au sein des grandes entreprises pour des tâches telles que les investissements, la publicité [Alt80]. Le terme "business intelligence" est utilisé pour décrire des systèmes décisionnels afin d'améliorer le management de l'entreprise. Ces techniques reposent généralement sur l'exploitation d'importantes et nombreuses bases de données.

Dans le domaine de la recherche opérationnelle, les techniques classiques d'aide à la décision utilisées efficacement pour les systèmes de petite taille, ont laissé place aux méthodes à critères multiples lorsqu'il s'agit de problématiques plus complexes comme la gestion des ressources, l'éducation, l'investissement, la défense ou encore la santé [KSS$^+$17]. Bernard Roy poursuit ses recherches sur le même thème [Roy90] et développe plusieurs versions de sa méthode Electre. Cette méthode consiste à rechercher une solution compromis qui peut prendre diverses formes : choix, affectation ou classement. Elle passe difficilement dans la communauté à ses débuts mais Bernard Roy est convaincu de la nécessité de prendre en considération l'ambiguïté, dans un premier temps, puis l'imprécision, l'incertitude, le "mal déterminé". Ses recherches le mèneront vers la fin de sa carrière, dans les années 2000, à explorer les outils d'aide à la décision pour des décisions collectives ou l'aide à la concertation [DR05].

La théorie de possibilités, qui étend les travaux sur les ensembles flous, propose fin des années 80 une représentation différente des probabilités. Ces dernières ont souvent été remises en question pour leur difficulté d'acquisition lorsque peu de données existent sur le problème à modéliser. Dubois et Prade [DP95] proposent une vision alternative des préférences à l'aide de la théorie des possibilités qui offrent un cadre plus souple, en particulier pour la représentation de l'incertitude et des risques. Le modèle de décision est qualitatif et de nature possibiliste ; les décisions peuvent être ordonnées selon deux critères qualitatifs, un pessimiste et un optimiste. Fin des années 90, le domaine de l'Intelligence Artificielle voit apparaître un nouveau paradigme d'aide à la décision appelé *théorie de la décision qualitative*

posée dans les papiers de Brafman et Tennenholtz [BT96] et de Doyle et Thomason [DT99]. Les décideurs ou experts ne sont pas toujours capables de définir l'incertitude de leurs processus sous la forme de distributions de probabilités, ni même de quantifier leurs préférences [Bou94]. Une représentation qualitative du problème de décision peut s'avérer suffisante pour répondre à un problème de décision, un modèle très précis n'étant en effet pas toujours exploité dans toute son ampleur pour répondre aux requêtes des décideurs. Formaliser un modèle de manière symbolique peut être plus adapté à l'expression de l'incertitude et des préférences. Des travaux récents ont exploré ce domaine de la décision en proposant des règles qualitatives dans l'incertain facilement interprétables, implémentables et ne nécessitant que peu d'information en entrée. Aujourd'hui ces méthodes s'intéresse à la prise de décision collectives dans un univers incertain [BAEF15].

Si la puissance de calcul, les algorithmes ont progressé de manière fulgurante durant ces dernières décennies, la nature particulière du problème de l'aide à la décision, en lien direct avec l'humain, complexe dans ses raisonnements et ses décisions, a constitué et constitue encore un problème de recherche difficile. La multitudes d'articles portant sur le sujet depuis les années 2000 montrent à quel point le sujet est dynamique. Nous allons nous intéresser, dans la section suivante, à la définition de l'aide à la décision, un mot souvent employé mais qui reste malgré tout assez vague, et comment y connecter tous les autres termes qui gravitent autour du sujet et que l'on retrouve dans de nombreuses publications.

1.2 Quelques définitions

L'aide à la décision a donné lieu à de nombreuses théories, des travaux de Simon à aujourd'hui. Pour commencer, nous présentons provisoirement une vision classique de l'aide à la décision utilisée surtout par les approches de type recherche opérationnelle.

Aide à la décision.
L'aide à la décision s'appuie sur des *modèles* pour aider un *acteur* à choisir la prochaine action parmi plusieurs alternatives.

Une démarche d'aide à la décision n'a de sens que si elle est mise en œuvre pour le compte d'un acteur. Dans les pays anglo-saxons, ce rôle est identifié par un terme particulier, le "stakeholder" dont la définition de référence est celle de Freeman [Fre84] : *"any group or individual who can affect or is affected by the achievement of the organization objectives"*. Si cette définition est originellement dédiée au management des organisations, elle peut tout à fait s'appliquer aux

problèmes de décision où intervient un système informatique. De son coté, Bernard Roy [RB93] sépare les "acteurs" en deux catégories, les "agis" aidés par les modèles dans leur prise de décision, et les "hommes d'études" ou "intervenants" qui conçoivent ces modèles en "fonction du système de valeurs dont ils sont porteurs". Cette distinction peut faire echo à celle du "stakeholder", les "agis" étant affectés par les conséquences de la décision quand les "hommes d'études" ont un réel impact sur elle, même si en théorie ceux-ci doivent proposer des éléments de réponses "en toute objectivité et neutralité", ce qui peut faire objet de discussions (voir à ce sujet [Roy93]). Dans la suite de ce manuscrit, nous utiliserons le terme "acteur" pour désigner les personnes pour le compte desquelles le système est effectué et qui participent activement au processus de décision.

Acteur.
Un acteur est un individu ou un ensemble d'individus pouvant affecter ou être affecté par le problème de décision.

L'aide à décision repose sur un modèle ou un type de modèle qui diffère selon les théories comme nous le verrons plus loin. Sans même parler des différents types de formalismes potentiels malgré notre intérêt pour ce sujet en informatique, il s'agit tout d'abord de s'accorder sur la nature du modèle. La définition du modèle peut-être celle donnée par Bernard Roy [Roy85] qui reste suffisamment généraliste pour fédérer les nombreux cas de figure.

Modèle (Bernard Roy)
Un modèle est un schéma qui, pour un champ de questions, est pris comme représentation d'une classe de phénomènes plus ou moins habilement dégagés de leur contexte par un observateur pour servir de support à l'investigation et/ou à la communication.

Pour Pomerol [Pom92], *"c'est le système (homme + artefact) qui forme le modèle de la tâche, l'artefact seul est inachevé... il faudra s'efforcer de dire quelle est la part du modèle et quelle est la part non modélisée, comment l'homme vit en symbiose avec le système et comment il complète le modèle toute en restant décideur libre"*.

Dans la théorie de la décision, on dénote quatre grands mouvements : la théorie normative, la théorie descriptive, la théorie prescriptive et la théorie constructive. Les frontières séparant ces différentes théories sont loin d'être franches puisqu'elles proviennent de l'appréciation du modèle formel employé. Les distinctions entre ces

différentes théories dépend du degré de *rationnalité* du modèle, de son processus d'acquisition et de l'interprétation de ses réponses.

théorie normative
Les théories normatives doivent répondre à la question : "Que doit faire l'acteur ?". Elles s'appuient sur des modèles précis construits à partir d'informations fiables n'intégrant aucun facteurs psychologiques. Les modèles développés sont très rationnels nécessitant un comportement semblable de la part des acteurs.

théorie descriptive
Les théories descriptives de leur coté sont plus réalistes et intègrent de nombreux facteurs humains afin de pouvoir répondre à la question "Que ferait vraiment un décideur (en situation réelle) ?". Les modèles sont construites à partir de l'observation des comportements. Ils sont plus généralistes et peuvent s'appliquer à une classe de problèmes d'aide à la décision de nature similaire.

théorie prescriptive
Les théories prescriptives peuvent être considérées comme une combinaison des deux premières approches alliant les fortes bases théoriques de l'approche normative avec l'intérêt de la représentation des facteurs humains qui influencent les décisions lorsqu'il s'agit aborder des situations complexes.

théorie constructive
Les théories constructives proposent au décideur de participer à la construction du modèle rationnel. Cette participation du décideur peut s'étendre au processus de décision lui-même dans sa globalité avec l'adjonction possible de ses préférences, ce qui complique le processus de production des recommandations pertinentes et personnalisées. Ces approches ont autant pour objectif de structurer et de formaliser le problème lui-même que sa résolution. Le décideur apprend à comprendre son problème au fur et à mesure qu'il le formalise.

De nombreux débats ont portés sur les distinctions et les frontières séparant les théories normatives et descriptives. Elles différent essentiellement par le type d'information intégrées dans le modèle qui, pour ce qui concerne les approches descriptives s'appuient sur la psychologie, la perception ou l'attitude de l'acteur, l'imprécision et l'incertitude de l'environnement de décision. En pratique, on constate que peu de théories de la décision sont purement normatives ou descriptives d'où l'émergence, grâce aux approches pluridisciplinaires, de la théorie prescriptive. Un exemple intéressant, en lien avec les travaux présentés dans ce mémoire, est le cas des modèles discrets employés dans l'analyse de décisions

économiques qui reposent à la fois sur des attributs représentant le système et sur le comportement du décideur [APKH16]. L'approche constructive défendue par Bouyssou et al. [BDPP09, BMPV06] est fortement connectée au *processus de décision*.

La définition de l'aide à la décision donnée par Bernard Roy [Roy85] est une définition complexe qui s'inscrit également dans la théorie des approches constructivistes.

Aide à la décision (Bernard Roy)
L'aide à la décision est l'activité de celui qui, prenant appui sur des modèles clairement explicités mais non nécessairement complètement formalisés, aide à obtenir des éléments de réponse aux questions que se pose un intervenant dans un processus de décision, éléments concourant à éclairer la décision et normalement à prescrire, ou simplement à favoriser, un comportement de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et le système de valeurs au service desquels cet intervenant se trouve placé d'autre part.

Si le concept de *processus d'aide à la décision* apparait dans cette définition, il est bien dû à Simon (1947) qui constatait, contrairement aux idées reçues à l'époque, que les décideurs n'avaient pas vraiment une idée précise de leur problème et que leur formulation pouvait souvent s'analyser comme une recherche de compromis. Intégrer les activités mentales du décideur dans un processus permet alors de construire avec lui un modèle intégrant les comportements liés à sa décision. Dans ce cadre, on ne considère plus que le décideur prend une décision unique et ponctuelle parmi un ensemble d'alternatives mais suit un processus mental durant lequel il devra effectuer une suite d'actions jusqu'à atteindre son but en fonction de ses priorités. Bouyssou et al. [BDPP09] proposent de définir le processus d'aide à la décision selon quatre niveaux. Ce processus ne suit pas nécessairement une construction linéaire.

Processus d'aide à la decision (Bouyssou et al.)
Le processus d'aide à la décision peut se représenter selon les quatre artéfacts cognitifs suivants :
- Représentation du problème
- Formulation du problème
- Modèle d'évaluation
- Recommandation finale

Afin d'accompagner la prise de décision, des systèmes informatiques ont été développés afin de fournir au décideur les informations nécessaires dans les meilleurs délais. Ils ne sont pas toujours identifiés comme tels, mais la plupart de ces systèmes peuvent se regrouper dans le terme de Systèmes Interactifs d'Aide à la Décision (SIAD) ou Decision Support Systems (DSS) en anglais. Même si en 1987, Keen [Kee87] se posait encore la question : *"What is the definition of DSS and does it matter that after over ten years of sustained research and practice there is still no established definition ?"*, la définition qui est toujours d'actualité est celle de Keen et Scott-Morton [KSM78].

Système Interactifs d'Aide à la Décision (Keen et Scott-Morton)
Les Système Interactifs d'Aide à la Décision SIAD impliquent l'utilisation d'ordinateurs pour :

- Assister les décideurs dans leur processus de décision dans des tâches semi-structurées ;
- Aider plutôt que remplacer le jugement des décideurs ;
- Améliorer la qualité de la prise de décision plutôt que l'efficacité.

La nature semi-structurée des situations nécessite un équilibre de la décision entre l'humain et les traitements informatiques. Si toute la décision pouvait être automatisée celle-ci n'aurait plus lieu d'être. L'interactivité est une caractéristique essentielle de ces systèmes. Pomerol [LJC95] interprète le fonctionnement du SIAD comme une *recherche heuristique* menée par le décideur dans un espace d'états qui contient aussi bien des modèles que des données. Le décideur doté de ses connaissances pratiques possède un "méta-modèle" du processus du décision et peut s'appuyer sur le SIAD appréciable pour ses capacités de traitement [LJC89]. Le processus de recherche heuristique qui mène à des éléments de solution du problème de décision peut se poursuivre jusqu'à ce que le système fournisse une solution acceptable pour le décideur. La qualité d'un SIAD et son acceptation auprès des décideurs, sera donc jugée sur sa capacité de dialogue, ses multiples perspectives d'analyse mais surtout son aptitude à expliquer le raisonnement ayant conduit à la solution. Dans ce contexte de SIAD laissant la part belle à l'acteur dans la conduite du processus de décision, dans ses choix (poursuite de la recherche heuristique, retour en arrière, nouvelle résolution), il est assez naturel d'avoir vu l'Intelligence Artificielle prendre une part de plus en plus prépondérante. Il est intéressant de remarquer avec le recul que toutes ces questions sont, à peu de choses près, les mêmes que celles qui se posent aujourd'hui avec les outils de fouille de données dans ce que l'on appelle *l'interprétabilité* et que je développerai dans les perspectives de ce mémoire.

1.3 L'Intelligence Artificielle pour l'aide à la décision

La théorie de la décision et l'Intelligence Artificielle se recoupent à de nombreux égards dans les formalismes, les approches et les communautés. Le processus d'aide à la décision proposé par Simon a par exemple été revisité selon un point de vue "raisonner pour décider" par Pomerol [Pom97]. Les approches proposées par l'IA relèvent pour la plupart des théories descriptives et proposent, en pratique, des solutions qui peuvent traiter :

— l'incertitude et l'ambiguïté des informations ;
— la prise en compte du raisonnement des acteurs ;
— l'intégration des préférences ;
— le volume des données ;
— la prise en compte des événements futurs ;
— une mesure de prévisibilité des événements agissant sur le système ;
— la nécessité d'expliquer et de commenter les solutions proposées.

Bien que l'aide à la décision soit une problématique relativement généralisée au sein de l'IA, ce n'est pas une thématique en tant que telle et elle se révèle dans de nombreux domaines selon les modélisations proposées (numériques ou symboliques). La figure 1.1 propose ma vision, schématique et sans doute subjective, des domaines de l'IA qui relèvent de l'aide à la décision pour l'humain (et non les machines). Les thématiques choisies sont celles présentées par Russel et Novig dans leur livre de référence "AI, A Modern Approach" [RN10]. En réalité, les thèmes ne sont pas si cloisonnés, de nombreuses autres thématiques existent ou celles présentées peuvent s'associer (comme les probabilités et la planification qui ici ont, par exemple, été regroupées dans le "raisonnement probabiliste" pour ce cas de figure). Les thématiques en bleu sont celles gérant au mieux le volume des données, celles en violet prennent le plus en compte la complexité des connaissances comme l'équivocalité qui renvoie à la présence de plusieurs interprétations simultanées mais divergentes d'un domaine décisionnel.

On peut distinguer deux types d'approches : les approches "axiomatiques" et les approches "réalistes". Les approches "axiomatiques" proposent de formaliser les processus mis en œuvre dans la décision comme la définition des règles de décision, la théorie des préférences, les procédures d'agrégation, etc. Les approches "réalistes" s'intéressent à une classe de problèmes, étayée par une application concrète, afin de fournir des méthodes ainsi qu'un processus d'utilisation aboutissant à un système informatique pouvant être testé par les décideurs. Bien entendu, ces approches ne fonctionnent pas non plus en vase clos, les approches "réalistes" peuvent valider les théories des approches "axiomatiques" celles-ci s'inspirant des expériences concrètes pour identifier les besoins en nouvelles théories.

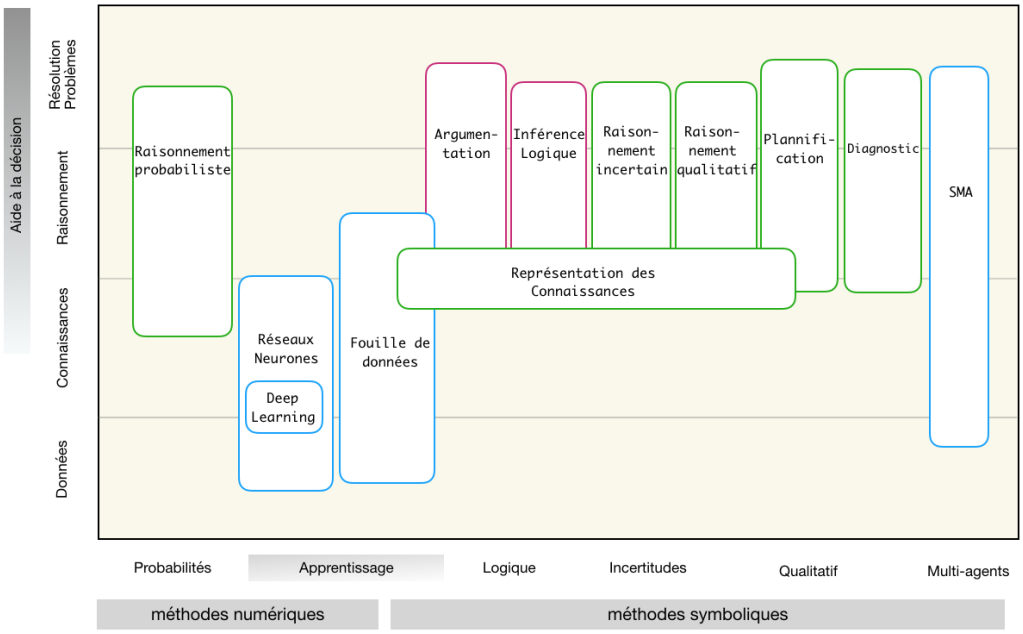


FIGURE 1.1 – Les domaines de l'IA et l'aide à la décision

Les domaines concernés se répartissent en majeure partie dans les thématiques qui appartiennent aux domaines de la *représentation des connaissances* et du *raisonnement*. Certaines approches appelées "modélisations hybrides" peuvent éventuellement associer des informations numériques aux connaissances. De nombreux formalismes existent pour représenter une situation complexe nécessitant une assistance à la décision. On choisira la représentation la plus appropriée pour représenter le système en fonction de la prédominance de ses caractéristiques et de sa taille. Cependant, le choix de la modélisation est aussi fortement lié à son type d'exploration et aux exigences d'efficacité requises pour fournir les éléments d'aide à la décision.

La première motivation de l'approche qualitative, en relation avec les travaux présentés dans ce mémoire, est l'efficacité espérée par la simplicité des modèles et la facilité d'acquisition des connaissances auprès des experts [BT96]. La première distinction concerne la gestion de l'incertitude qui remplace les probabilités des approches classiques par des valeurs non numériques, éventuellement plus abstraites, mais facilitant leur acquisition auprès des experts. Par exemple, sur un problème de diagnostic, Darwiche et Goldszmidt [DG94] proposent une utilisation du calcul Kappa, une version qualitative du raisonnement bayésien, en élicitant les informations à l'aide de règles et en spécifiant l'ignorance possible sur un

événement. D'autres variantes qualitatives de la théorie de la décision ont été proposées [Bou94, Pea93, DFP03] dans lesquelles les préférences sont incertaines et formulées par des termes logiques, éventuellement sujets à la révision. S'appuyant sur ce précédent domaine, Pomerol [Pom97] a proposé un processus de décision réaliste en deux étapes : i) le diagnostic, ii) la projection vers l'avenir ("look-ahead" en anglais). Il s'agit dans ce schéma de distinguer les états courants et les états futurs dans l'objectif de pouvoir prédire l'avenir en chaînant les événements et les actes, dans ce que l'on peut appeler "scenarios". Le comportement d'anticipation reproduisant la capacité de l'homme à se placer dans d'hypothétiques situations futures, sera alors reproduit par l'analyse "what-if". Les modèles de simulation apparaissent comme un moyen d'effectuer cette analyse malgré le frein constitué par les boucles de rétroaction. De nombreux formalismes (pour la plupart issus des graphes) peuvent être utilisés en fonction du type de données, du degré d'incertitude ou de la granularité temporelle. Leur efficacité dépend du type de problème à modéliser et des algorithmes ou recherches heuristiques mis en œuvre pour leur exploration.

Emanant du raisonnement qualitatif, l'argumentation, qui est un champ disciplinaire récent de l'Intelligence Artificielle, traite de manière originale l'aide à la décision en s'intéressant aux arguments, éventuellement conflictuels, avancés en faveur ou non d'une situation complexe, d'une action. L'intérêt de cette approche est de s'appuyer sur un raisonnement composé d'étapes modulaires et intuitives. Jusqu'à la fin des années 2000, l'argumentation a surtout été utilisée pour la détection des inconsistances dans les bases de données [AC09, BH01] et la modélisation des différentes formes de dialogue comme la persuasion [Amg00] ou la négociation. Le premier framework basé sur l'argumentation pour l'aide à la décision a été proposé par Amgoud et Prade [AP09]. Ce framework traite différents types de problèmes d'aide à la décision pouvant comporter de l'incertitude, des critères multiples de décision ou des décisions à base de règles. Il distingue deux types d'arguments, les premiers appelés "arguments épistémiques " qui justifient des croyances (par exemple "le temps qu'il va faire" fourni par les modèles météorologiques) et les "arguments pratiques" qui justifient des options (comme celle de prendre un parapluie s'il pleut). Ces travaux ont été à la base d'autres études comme celles reliant les explications à l'argumentation (voir Ferretti et al. [FTG$^+$17] pour un récent état de l'art ou Garcia et al. [GCRS13] pour la formalisation des "explications dialectiques"). L'intégration du raisonnement à base d'argumentation a donné lieu à de nombreux systèmes de recommandations dans des domaines variés comme les suggestions de choix de films [BBD$^+$14], les recommandations en matière d'alimentation [BTMJ13] ou l'aide à la prise de décisions médicales [CTC$^+$12].

Dans un registre plus technique, où les connaissances associationnistes employées dans les précédentes approches sont remplacées par la notion de "modèle"

décrivant la structure d'un système, le diagnostic est une branche à part entière de l'IA. Son objectif est d'établir l'état d'un système partir d'un modèle de comportement et d'observations reçues sur ce système [Rei87, LZ03]. Nous détaillerons cette thématique dans le prochain chapitre. La planification peut se rapprocher du diagnostic dans certaines de ses approches par la modélisation du système étudié. Son objectif consiste à sélectionner et à ordonnancer des actions permettant d'atteindre un but défini. Si la planification est grandement utilisée dans les systèmes intelligents (robotique), elle a été également très employée pour l'élaboration de systèmes d'aide à la décision. Des exemples de diverses approches de planification pour la décision dans les agro-écosystèmes sont par exemple donnés par Quesnel et al. [QAB$^+$13]. Le problème de planification simple concerne un environnement déterministe qui peut être complètement observé, mais la plupart des systèmes réalistes sont complexes avec, par exemple, des environnements partiellement observables ou de l'incertitude sur les effets des actions. Les modes de représentations classiques de la planification sont la logique, les graphes d'états, la satisfaction des contraintes (CSP), les langages de formalisation des actions. Pour des environnements plus complexes, les approches reposent sur des systèmes d'états-transitions comme les automates ou les réseaux de Pétri ou des structures de types graphes comme les processus de décision markoviens qui définissent des probabilités pour représenter l'incertitude des actions.

Enfin, une dernière thématique, même si elle est assez éloignée de notre approche, ne peut être passée sous silence tant elle propose de nombreux outils d'aide la décision dans tous les domaines, et en particulier en agro-écologie [BP04]. Il s'agit des systèmes multi-agents. Dans ce domaine, il existe une branche appelée *Multi-Agent Decision Making* [Bul14] qui, contrairement aux approches classiques, propose que les actions et les décisions des agents soient interdépendantes, ce qui peut éventuellement mener à des buts inconsistants. Ce principe repose sur un processus interactif mis en œuvre par des protocoles de communication entre agents jusqu'à l'obtention d'un accord. On retrouve également de nombreuses techniques pour la négociation reposant sur des modèles coopératifs, de type axiomatiques, ou non-coopératifs, généralement procéduraux, la théorie des jeux ou l'argumentation. S'il existe de nombreuses applications pratiques pour traiter les problèmes, les formalismes employés pour représenter les systèmes complexes sont les "jeux stochastiques" tels que les processus de décision markoviens partiellement observables et interprétables [GP05, PG17], ou les modèles purement qualitatifs combinés à des logiques spécifiques.

Ce tour d'horizon ne saurait être exhaustif, tant les méthodes et applications peuvent être variées. Ce chapitre propose *une* interprétation de l'aide à la décision qui comme nous avons pu le voir peut tout à la fois se traiter de manière très

théorique ou appliquée. Elle relève aussi bien de la recherche opérationnelle, des sciences cognitives ou de l'intelligence d'entreprise (non abordées), de la théorie de la décision, présentée ici dans les grandes lignes car pouvant devenir spécialisée et très formelle, et des nombreux domaines de l'IA qui tentent de résoudre de manière computationnelle ce problème. Parmi toutes ces approches possibles, il a été nécessaire de faire des choix. Ces choix ont été conditionnés par les questions suivantes. i) *Quel type de système est considéré par nos études ?* Nos thématiques et nos applications nous ont porté vers les systèmes dynamiques complexes ayant une dynamique temporelle. ii) *Quelle est l'aide apportée aux décideurs ?* Fournir une explication sur le raisonnement mise en œuvre sera une priorité dans les travaux développés.

Chapitre 2

Diagnostic de systèmes à événements discrets

Sommaire

2.1 Définitions et travaux de référence 30
2.1.1 Des modèles pour expliquer 30
2.1.2 Terminologie 31
2.1.3 Systèmes à événements discrets 33
2.1.4 Définitions formelles et originales du diagnostic de SED 35
2.1.5 Approches décentralisée et distribuée 39
2.2 Diagnostic incrémental et décentralisé 40
2.2.1 Représentation 41
2.2.2 Diagnostic incrémental 43
2.2.3 Diagnostic décentralisé 44
2.2.4 Diagnostic de systèmes reconfigurables 45
2.3 Autres méthodes de diagnostic 46
2.3.1 Diagnostic par model-checking 46
2.3.2 Diagnostic avec ASP 55

La première étape du processus de décision est celle du diagnostic [Pom97]. Le problème du diagnostic consiste à déterminer avec la plus grande précision possible l'état courant d'un système tout en tenant compte de son passé. Ce chapitre présente le diagnostic de systèmes dynamiques représentés par les *systèmes à événements discrets (SED).*

Une première partie décrit le domaine du diagnostic à base de modèles et les travaux précurseurs à l'aide des SED. Ce contexte permet de situer mes contribu-

tions. La seconde partie aborde un travail réalisé dans le cadre d'une thèse, que j'ai co-encadrée, et qui porte sur le diagnostic incrémental et décentralisé pouvant s'appliquer aux systèmes reconfigurables. Je me suis également intéressée au diagnostic à l'aide de méthodes formelles ou logiques qui constituent une alternative pour explorer et expliquer les modèles. Ces approches sont présentées dans la dernière partie de ce chapitre.

2.1 Définitions et travaux de référence

Le diagnostic est un domaine de l'Intelligence Artificielle qui vise à détecter, isoler et identifier les dysfonctionnements d'un système. Le principe du diagnostic consiste à s'appuyer sur des *observations* pour inférer les comportements anormaux (encore appelés *pannes*) sachant que ces observations sont parfois imprécises, incertaines voire fausses. Les premières approches développées pour le diagnostic se sont appuyées sur les systèmes experts dans les années 70 [Dav82]. Cependant ils ont rapidement montré leurs limites en particulier lorsque les systèmes à étudier étaient d'une nature autre que médicale [Par87]. Les chercheurs se sont alors concentrés sur une modélisation à partir, non plus du savoir expert, mais des connaissances sur le comportement du système. Le diagnostic s'appuie sur un modèle comportemental et l'on parle de *diagnostic à base de modèles* (*model-based diagnosis* en anglais)

2.1.1 Des modèles pour expliquer

La première théorie du diagnostic à base de modèles, présentée par Reiter [Rei87] et de Kleer et Williams [JW87] à la fin des années 80, repose sur les principes suivants :

- un *artefact* (système) à diagnostiquer dans un contexte réaliste,
- une théorie logique pour représenter la structure et le comportement de l'artefact,
- des observations obtenues depuis le contexte et associées dans la théorie logique,
- un algorithme pour calculer le diagnostic.

Dans ce cadre, Reiter, de Kleer et Williams proposent que l'*explication* fournie pour le diagnostic soit tout état du système qui satisfait à la fois le modèle (la représentation de l'artéfact) et les observations collectées. À la suite de ces recherches, on peut identifier deux catégories de raisonnement non monotones pour la recherche du diagnostic : le diagnostic basé sur la consistance (*consistency-based diagnosis*) et le diagnostic abductif. Le raisonnement basé sur la consistance suit

l'approche initiée par Reiter, De Kleer et Williams selon laquelle le modèle est utilisé pour prédire l'état du système et le diagnostic obtenu par comparaison des prédictions avec les observations recueillies sur le système (cf. Figure 2.1). Le modèle est conçu pour décrire le comportement du système lorsqu'il fonctionne normalement. L'explication apportée par le diagnostic consiste à déterminer la partie du système à l'origine du comportement imprévu dans le but de pouvoir agir sur le système.

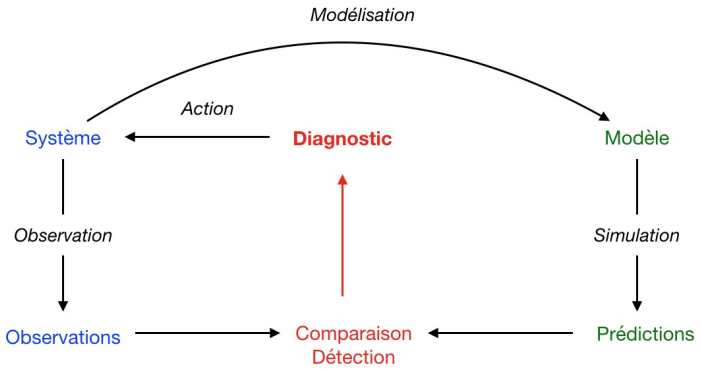


FIGURE 2.1 – Principe du diagnostic à base de modèle (consistency-based diagnosis)

Le raisonnement abductif a pour origine le raisonnement médical au sein duquel on explique généralement un symptôme en cherchant l'ensemble des causes du symptôme. Une hypothèse s'apparente à un symptôme qui explique de manière logique son apparition. La représentation utilise un *modèle causal* qui contient une information explicite sur les fautes et leurs conséquences (éventuellement en cascade) jusqu'à la manifestation des événements observés [CP87].

2.1.2 Terminologie

Les définitions visent à poser de manière claire et précise les termes employés dans le domaine du diagnostic. Quels que soient les domaines traités, systèmes industriels, médecine, télécommunications, etc. les approches s'appuient sur une terminologie commune même si les techniques développées peuvent être différentes.

Le centre d'intérêt du "diagnostic" est le *système*. Ce système, considéré comme faillible, est observé, supervisé afin de détecter s'il fonctionne correctement ou non.

Système
Un système est une entité composée d'éléments (les composants) qui évolue dans le temps avec pour but de remplir une fonction définie. Les éléments sont interconnectés et échangent de la matière, de l'énergie ou des informations.

La philosophie du diagnostic est de représenter le système par un modèle si possible réutilisable, pour d'autres types de tâches que le diagnostic (supervision, planification) ou pour d'autres systèmes comparables.

Modèle
Un modèle est une abstraction du système dans un formalisme permettant de décrire sa *structure* et de simuler son *comportement*.

Cette définition précise pour le cadre du diagnostic la définition du modèle, plus générale, posée pour l'aide à la décision donnée par Roy. Un modèle peut être quantitatif ou qualitatif (en utilisant différents types d'abstraction qualitative), dynamique ou statique (la dynamique pouvant avoir subi une abstraction par soucis de simplicité), et peut être éventuellement probabiliste ou utiliser différents moyens de représentation de l'incertitude ou de l'imprécision.

La représentation à base de composants est la plus répandue actuellement mais pas la seule puisque, dans certains systèmes, il peut être difficile d'identifier des composants et le comportement global du système ne correspond pas nécessairement à la composition des comportements de ses composants. Il est possible de concevoir un modèle de *bon fonctionnement* ou un modèle intégrant les dysfonctionnements, selon le type de raisonnement appliqué pour le diagnostic.

Composant
Un composant est un élément du système (sous-système) interconnecté avec d'autres composants. C'est l'unité minimale de description du système.

Les observations sont indispensables au problème du diagnostic puisqu'elles seules permettent d'évaluer si le système a un comportement normal ou non [Rei87].

Observation
Une observation est une information obtenue sur le comportement du système. Elle peut être recueillie en entrée, en sortie ou sur un composant.

Les observations sont généralement émises par des capteurs chargés de surveiller le système (qui peuvent ne pas être intégrés à la modélisation car extérieurs au système) mais dans la plupart des approches, et par abus de langage, on parle d'observations émises par le système. Il arrive cependant que les observations reçues

soient différentes des observations émises et il est alors nécessaire de procéder à une reconstruction des observations émises. On parle de *complétude* lorsque l'ensemble des observations reçues est le même que l'ensemble des observations émises. Les observations peuvent être datées mais les problèmes de synchronisation des horloges des capteurs et les délais de transmissions peuvent occasionner un problème d'ordre des observations.

Panne
Une panne (*fault* en anglais) correspond à au moins une déviation du système par rapport à son comportement normal, acceptable ou nominal.

De nombreux termes sont associés à la notion de panne : la *détection de panne* permet de déterminer si une faute est présente et à quel temps elle s'est produite ; l'*isolation de la panne* identifie le composant mis en cause.

Défaillance
Une défaillance décrit une incapacité du système à remplir sa fonction

Afin de pouvoir établir le diagnostic d'un système, il peut être utile de savoir si celui-ci est *diagnosticable* [SSL+96].

Diagnosticabilité
La diagnosticabilité est la capacité de mesurer si, connaissant un système et un ensemble d'observations, il est possible de diagnostiquer les pannes dans un délai fini.

La diagnosticabilité permet d'être assuré de récupérer un diagnostic en un temps borné après l'occurrence de la panne et de connaître celles qui ne peuvent être détectées par le diagnostic.

2.1.3 Systèmes à événements discrets

Les systèmes à événements discrets présentent plusieurs avantages pour la modélisation et le diagnostic de systèmes dynamiques. Cette représentation introduite par Sampath, Lafortune et al. [SSL+96] a été largement étudiée dans le monde du diagnostic (le lecteur intéressé trouvera un état de l'art récent écrit par Zaytoon et Lafortune [ZL13]). Cassandra et Lafortune [CL08] définissent un système dynamique de la manière suivante :

Système Dynamique (Cassandra et Lafortune)
Un système dynamique est un système dont la sortie dépend des précédentes valeurs en entrée.

Alors que dans un système statique, la sortie est indépendante des précédentes valeurs en entrée, il est nécessaire, pour un système dynamique, de conserver un historique de ces valeurs d'entrée. Tout ce qui est extérieur au système est appelé *environnement* du système.

Généralement, on classe les systèmes dynamiques en deux catégories : les *systèmes continus* dont la représentation suit le comportement physique habituellement à l'aide d'équations différentielles et *les systèmes à événements discrets* (SED) qui expriment la logique du comportement. Les SED se caractérisent par les deux propriétés suivantes : i) l'état est décrit par des grandeurs discrètes, ii) la dynamique du système est conduite par l'occurrence d'événements à certains instants.

Dans ce cadre, un système est décrit par un ensemble de variables. Soit $u(t)$ le sous-ensemble des variables en entrée complètement défini pour tout $t \geq t_0$ et $y(t)$, le sous-ensemble des variables en sortie qu'il est possible de mesurer. L'*état* d'un système à un instant t doit décrire son comportement à cet instant, et de manière mesurable. La définition d'un état donnée par Lafortune [CL08] est la suivante :

ETAT
L'*état* d'un système dynamique au temps t_0 est l'information requise à t_0 telle que la sortie $y(t)$, pour tous les $t \geq t_0$ est uniquement déterminée à partir de cette information et de $u(t)$, $t \geq t_0$

Dans les SED, l'espace d'états est un ensemble discret. La dynamique du comportement d'un système discret est en conséquence plus intuitive puisqu'il s'agit de passer selon un raisonnement logique d'un état à un autre état. Par exemple, une ampoule dispose d'un ensemble de deux états, *éteint* et *allumé*, et peut passer de l'un à l'autre en fonction de l'arrivée ou non du courant.

La notion d'*événement*, le moteur de la dynamique du système est défini de la manière suivante.

EVÉNEMENT
Un *événement* peut être une action, une manifestation extérieure ou une conjonction de conditions qui survient de manière *instantanée* et permet de faire passer le système d'un état à un autre état.

Dans notre exemple, deux événements sont possibles, l'*arrivée du courant* qui fait passer l'ampoule de *éteint* à *allumé* et *la coupure du courant* qui a l'effet inverse. Un événement qui provient de l'environnement du système est appelé un événement *exogène* alors que s'il provient du système lui-même, on parle d'un

événement *endogène*.

Le mécanisme temporel associé aux systèmes dynamiques est, en général, géré par une horloge qui mesure le temps et qui symbolise un "tic d'horloge". Selon l'apparition des événements vis-à-vis de ce tic d'horloge, on considère deux types de dynamiques qui peuvent être associées aux événements :

— A chaque tic d'horloge, un événement est susceptible de se produire. Dans le cas où aucun événement ne survient, l'état du système reste inchangé. Les transitions du système sont *synchronisées* sur les valeurs de l'horloge. On parle de système *time-driven*.

— Les événements peuvent survenir à différents instants, non prévus à l'avance et sans nécessairement coïncider avec le tic de l'horloge. Chaque événement est associé à son instant d'apparition. Les transitions du système résultent de la prise en compte de ces événements *asynchrones* et *concurrents*. On parle de système *event-driven*.

Lafortune et Cassandras [CL08] restreignent les SED aux systèmes *event-driven*.

Système à Événements Discrets (SED)
Un système à événements discrets est un système ayant un espace d'états discret, de type *event-driven* et dont l'évolution de l'état dépend entièrement de l'occurrence d'événements discrets asynchrones au cours du temps.

Dans de nombreux cas de figure, il n'est pas nécessaire de considérer la variable de temps si l'on peut associer à chaque transition entre états une occurrence d'événement. La trajectoire peut évoluer d'un état à un autre en fonction de l'occurrence d'un événement. On peut remarquer cependant que i) un même événement peut mener à des états différents du système, ii) deux événements différents peuvent conduire à un même état, iii) des événements dans certaines configurations ne provoquent pas nécessairement un changement d'état.

2.1.4 Définitions formelles et originales du diagnostic de SED

Si de nombreux travaux ont abordé le sujet du diagnostic de SED depuis une vingtaine d'années, ils reposent tous, à des variations près (compilation des pannes, outils de modélisation, types de pannes, architecture du système, etc.), sur la définition formelle donnée par Sampath et al. [SSL+95, SSL+96]. Dans ce cadre, la définition du système repose sur un modèle d'automate classique :

Système
Un système G est défini par $G = (X, E, \delta, x_0)$ tel que :
- X est l'ensemble des états du système (les états normaux et anormaux)
- $E = E_o \cup E_{uo}$ est l'ensemble des événements partitionnés entre les événements observables E_o et les événements non observables E_{uo}
- δ est la fonction de transition telle que $\delta : X \times E \to X$
- x_0 est l'état initial.

Les pannes sont généralement représentées à l'aide des événements non observables (sinon leur détection serait immédiate). Les événements non observables contiennent non seulement les fautes mais aussi les événements que les capteurs ne peuvent détecter (trop coûteux ou impossible à observer). Par la suite, on considère que E est un ensemble fini d'événements, chaque événement réel e sera toujours associé à une classe d'équivalence et le nombre de classes d'équivalence est fini.

Trace ou Trajectoire
Une trace du système est une séquence d'événements de E dont tout préfixe est lui-même une trace.

Une trace ou trajectoire permet de définir un comportement du système. Une séquence vide d'événements est également un comportement. À cette notion de trace peut être associée une opération de projection fournissant une trace plus abstraite. Un système est caractérisé par l'ensemble de ses comportements. À un instant t, une séquence finie d'événements définit le comportement du système.

Le problème du diagnostic consiste à déterminer, à partir du modèle du système, quelles sont les pannes (événements non observables) qui expliquent une séquence d'observations. Dans le cadre des SED, on considère qu'il s'agit de fournir la ou les traces du système à partir desquelles il est possible d'inférer les pannes.

Diagnostic de SED
Etant donnés un modèle du système et une séquence d'événements observables, le diagnostic du système consiste à calculer toutes les trajectoires sur le modèle qui permettent de générer les événements observés.

Le problème du diagnostic de SED peut s'appuyer sur la notion de *diagnostiqueur* pour le calcul du diagnostic hors-ligne [SSL+95, SSL+96]. L'approche repose sur la notion de diagnosticabilité qui permet d'assurer la détection de la faute.

DIAGNOSTICABILITÉ
Une panne est diagnosticable si elle peut être détectée avec certitude en un nombre fini d'observations après son occurrence. Une panne f est diagnosticable si, pour toute trace d'exécution s aboutissant à f, il existe une trace t suffisamment longue telle qu'il n'existe aucune autre trace d'exécution ne pouvant se distinguer de la trace $s.t$

Seuls les événements observables sont considérés comme éléments de la trace (on parle de projection sur les événements observables qui filtre tous les événements non observables). Cette définition indique que toute trace du système ayant la même séquence d'événements observables que $s.t$ contient la panne f. La figure 2.2 présente un exemple de trace $s.t$ dont le préfixe s se termine par une panne f et dont le suffixe t se termine dans l'état a. La projection $P(s.t)$ donne une suite d'événements observables. La panne f est *diagnosticable* si toutes les traces du système produisant la même suite d'événements que $P(s.t)$ contiennent également la panne f.

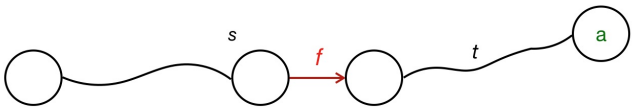


FIGURE 2.2 – Illustration de la diagnosticabilté

On dit qu'une panne est *n-observable* si elle peut être détectée avec un nombre spécifié d'observations n après son occurrence. Par extension on dira qu'un système est observable si toutes les occurrences de pannes peuvent être détectées dans un délai fini.

À partir de cette définition de la diagnosticabilité il est possible de construire un *diagnostiqueur* sur la base d'un automate *observeur*. Les états d'un *diagnostiqueur* sont associés à des labels F et/ou N : F si l'état peut être atteint par une trace contenant une panne f, N si l'état n'a pas été atteint par une trace contenant la panne. La figure 2.3 tirée de [ZL13] est l'exemple classique du diagnostiqueur (à droite) obtenu à partir de l'automate du système. Les événements non-observables f et u_o n'apparaissent pas dans la projection des traces possibles contrairement à a, b ou c qui sont observables. Les états du diagnostiqueur correspondent aux états atteints suite à l'apparition des observables. Ils peuvent être *normaux* comme $1N$ signifiant qu'aucune panne n'a pu se produire, *f-certain* comme $\{5F, 6F\}$ ou $\{4F\}$ indiquant avec certitude qu'une panne s'est produite et enfin *f-incertain* comme

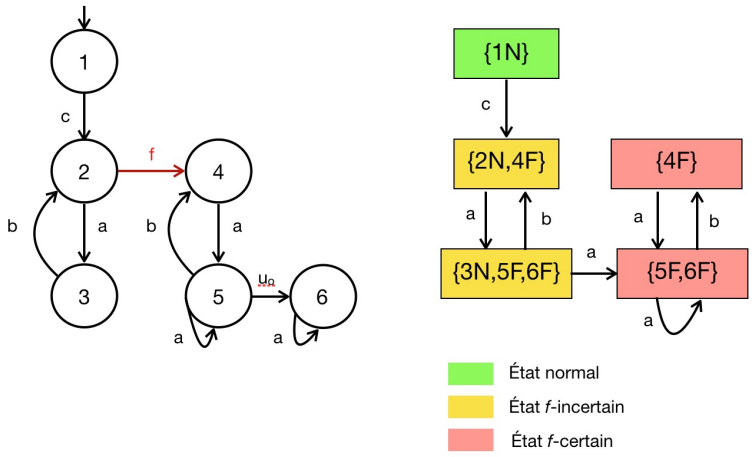


FIGURE 2.3 – Diagnostiqueur (à droite) établi à partir de l'automate modèle du système

$\{2N, 4F\}$,$\{3N, 5F, 6F\}$ qu'une faute a pu, ou non, se produire, en fonction des observations et donc du chemin emprunté par le système.

On appelle un *cycle f-indéterminé* un cycle dans le diagnostiqueur tel qu'il est composé essentiellement d'états incertains, comme dans l'exemple le cycle de trace $a.b$ qui relie les états $\{2N, 4F\}$ et $\{3N, 5F, 6F\}$. Ce cycle doit avoir pour particularité d'être composé de deux traces tel que que la faute f apparaisse dans la première mais pas dans la seconde. Sampath [SSL+95, SSL+96] démontre qu'un système est diagnosticable si et seulement si il n'existe pas de cycle f-indéterminé dans le diagnostiqueur pour chaque type de faute f.

Les précédentes définitions constituent les fondamentaux des problèmes de diagnostic et de diagnosticabilité des SED. Le diagnostiqueur est un outil efficace puisqu'il permet de représenter le problème du diagnostic à l'aide d'une représentation compacte où chaque état du diagnostiqueur est un diagnostic possible et où tous les diagnostics possibles apparaissent dans un état de cet automate. L'algorithme de diagnostic est efficace puisque la prise en compte d'une nouvelle observation a pour simple effet de franchir une transition. La contrepartie de cette approche réside dans la nécessité de disposer d'un modèle de fonctionnement du système correct et complet (difficile à obtenir pour les systèmes complexes), et dans la taille du modèle qui est exponentielle avec le nombre d'états et de pannes.

2.1.5 Approches décentralisée et distribuée

Afin de limiter la taille du modèle et la complexité, de nombreuses approches ont été étudiées depuis celle de Sampath. L'approche décentralisée offre une alternative intéressante lorsque le système est trop conséquent pour être traité par un seul modèle de diagnostic. Elle s'applique notamment aux systèmes au sein desquels l'information est décentralisée sur plusieurs entités (sous-systèmes, sites, etc.) ce qui est le cas de nombreux systèmes réels (réseaux de télécommunication, véhicules, chaînes de production). Le système est considéré comme un ensemble de sous-systèmes, appelés *composants*, ces composants étant interconnectés. Les algorithmes de diagnostic peuvent être de deux types : l'*approche décentralisée* où le calcul d'un diagnostic local est effectué au sein de chaque sous-partie du système avant d'être envoyé à un superviseur en charge de calculer le diagnostic global ; l'*approche distribuée* qui se passe d'un superviseur central et dans laquelle chaque sous-système effectue un diagnostic local et communique avec les autres sous-systèmes à travers un protocole de communication [DLT00]. Ces deux approches se rejoignent sur la modélisation par partie de chacun de ses composants. Le choix entre ces deux approches dépend essentiellement de l'usage requis et de l'architecture du système. Si le comportement global du système est nécessaire à son contrôle alors l'approche décentralisée sera privilégiée, si seul le fonctionnement local des sous-systèmes est nécessaire, l'approche distribuée sera utilisée.

Des premières approches de diagnostic décentralisé par SED ont été développées au sein de notre équipe DREAM à l'IRISA. Elles reposent sur le formalisme d'automates communicants décrivant chaque composant du système. Ces automates communiquent par le biais de *connexions* qui relient un port de sortie d'un composant au port d'entrée d'un autre composant permettant l'émission et la réception de messages synchrones. Afin d'éviter le modèle global, les travaux de Pencolé, Cordier et Rozé [PCR02] proposent de construire un *diagnostic local* au sein de chaque composant puis de fusionner les diagnostics locaux selon une certaine *stratégie* pour produire le diagnostic de pannes. Le diagnostic local est calculé en émettant des hypothèses sur la communication entre les composants. Ces hypothèses sont vérifiées lors de la fusion des diagnostics locaux (en vérifiant qu'un message émis est bien reçu) qui est effectuée deux à deux selon une stratégie optimisant l'ordre de fusion (en favorisant les composants en interaction) afin de limiter la taille du diagnostic obtenu.

2.2 Diagnostic incrémental et décentralisé

Mes premières contributions au domaine du diagnostic se situent dans le cadre posé ci-dessus. Je me suis intéressée à la réduction de la complexité pour le calcul du diagnostic. Ce travail a été réalisé durant le stage de Master 2 puis de la thèse d'Alban Grastien, que j'ai co-encadrés avec Marie-Odile Cordier.

Les premiers travaux ont porté sur la réduction du modèle à l'aide d'une propriété appelée *inversibilité*. Cette propriété a été définie sur les événements (ou actions) du modèle de système et est utilisée pour tailler l'espace de recherche sans perte d'information. Intuitivement, deux événements sont dits inversibles lorsque, après avoir transité par une trace comprenant ces deux événements, l'état atteint par le système est indépendant de l'ordre d'apparition de ces événements. Cette propriété fournit un moyen efficace de représenter des ensembles de trajectoires et est exploitée pour restreindre l'ensemble des comportements à prendre en compte dans le diagnostic ou même les systèmes de planification. Deux algorithmes sont proposés : le premier calcule efficacement les trajectoires et le second calcule automatiquement les propriétés d'inversibilité à partir des automates. La propriété d'inversibilité est liée à la propriété d'indépendance entre les événements utilisés dans la réduction de l'ordre partiel et à la notion de trace associée. Ces algorithmes ont été illustrés sur deux cas de figure en diagnostic et en plannification [GCLP03].

La thèse d'Alban Grastien fut consacrée à l'extension du diagnostic décentralisé au cas des systèmes *reconfigurables* en proposant une méthode de calcul *en ligne*. Cette approche s'appuie sur la représentation des observations sous forme de *chaînes d'automates* permettant le calcul incrémental du diagnostic. Les hypothèses posées pour ce travail élargissent le problème au cas d'observations d'ordre partiel pouvant comporter des incertitudes voire des pertes. Le calcul du diagnostic décentralisé par fusion tel que présenté par Pencolé, Cordier et Rozé [PCR02] n'empêche pas toujours l'explosion combinatoire du diagnostic global en particulier si la période d'observation est longue. Le diagnostic décentralisé peut tendre vers le diagnostic global et, dans le cas d'un usage en-ligne, la fusion se complexifie puisqu'elle doit alors traiter des diagnostics locaux de grande taille. L'approche que nous avons proposée diffère des précédentes dans le calcul du diagnostic local qui s'effectue de manière incrémentale dans une *fenêtre d'observation* [Gra05].

Problématique
Le principe du diagnostic consiste à synchroniser le comportement du système (qu'il soit décentralisé ou non) avec l'automate des observations. Le complexité augmentant avec la taille de l'automate des observations, la problématique traitée dans ce travail consiste à découper cet automate d'observations selon une fenêtre

temporelle afin de calculer des *diagnostics par morceaux*. On appelle *chaîne d'automates* ces automates modulaires qui permettent de représenter, ou des observations, ou des diagnostics locaux, correspondant à une fenêtre de temps. Nous avons montré que le diagnostic global pouvait alors être reconstruit à partir de la chaîne d'automates de diagnostic. L'intérêt de ce principe est de profiter du calcul du diagnostic de la fenêtre précédente pour calculer le diagnostic de la fenêtre courante. On parle alors de *diagnostic incrémental*.

2.2.1 Représentation

Le formalisme des *automates synchronisés* est utilisé pour représenter à la fois le comportement du système, les observations et le diagnostic. Une *trajectoire* dans un automate de comportement est une séquence d'états dont le premier est un état initial et le dernier un état final reliés par une suite de labels d'événements permettant le passage d'un état à l'état suivant. Les observations sont émises par des canaux de communication qui peuvent générer des anomalies ou des délais avec pour conséquence un ordre reçu différent de l'ordre émis, une incertitude sur l'événement ou une perte d'observations en cas de panne du capteur. Une représentation des événements observables sous la forme d'un automate, appelé *automate des observations* (bien que ce soient en réalité des événements) permet de représenter différentes séquences d'observations émises par le système à partir des observations reçues. Le diagnostic est obtenu par la synchronisation de l'automate, modèle de comportement, avec l'automate des observations.

Le calcul du diagnostic repose généralement sur les principes suivants :

— Le *modèle du système* noté MOD est un automate représentant l'ensemble des états possibles à t_0.
— Les *observations* pour lesquelles on ne dispose pas d'un ordre total sont également représentées par un automate noté OBS_n pour la période $[t_0, t_n]$.
— Le *diagnostic* noté Δ_n est un automate décrivant les trajectoires possibles sur le modèle du système compatibles avec les observations émises par le système durant la période $[t_0, t_n]$. Le diagnostic global du système est calculé par $\Delta_n = OBS_n \otimes MOD$ ($\otimes$ est une opération de synchronisation sur les labels des événements).

La méthode proposée s'appuie sur une fenêtre temporelle $[t_0, t_n]$ au sein de laquelle les diagnostics locaux sont calculés.

Diagnostic sur la fenêtre temporelle $[t_0, t_n]$

- $\mathcal{W}^i$ est une fenêtre temporelle correspondant à l'intervalle de temps $[t_{i-1}, t_i]$ avec $i \in [1, n]$,
- OBS_i est l'automate des observations pour $\mathcal{W}^i$,
- Δ_i est le diagnostic local pour $\mathcal{W}^i$,
- la *reconstruction* est l'opération qui permet de calculer le diagnostic global Δ_n à partir de la *chaîne d'automates* des diagnostics locaux Δ_i mais aussi de la séquence d'observations OBS_n à partir de la chaîne d'automates des observations OBS_i.

Chaîne d'automates

L'approche repose sur la définition d'une chaîne d'automates afin de représenter les observations et les diagnostics par morceaux. On considère qu'une séquence d'automates est une chaîne d'automates si un état final d'un automate est décrit comme un état initial de l'automate suivant et qu'en dehors de cette exception concernant les états *frontières*, un état ne peut appartenir qu'à un seul automate. La figure 2.4 présente une chaîne de trois automates (a) et l'automate global obtenu après reconstruction (b).

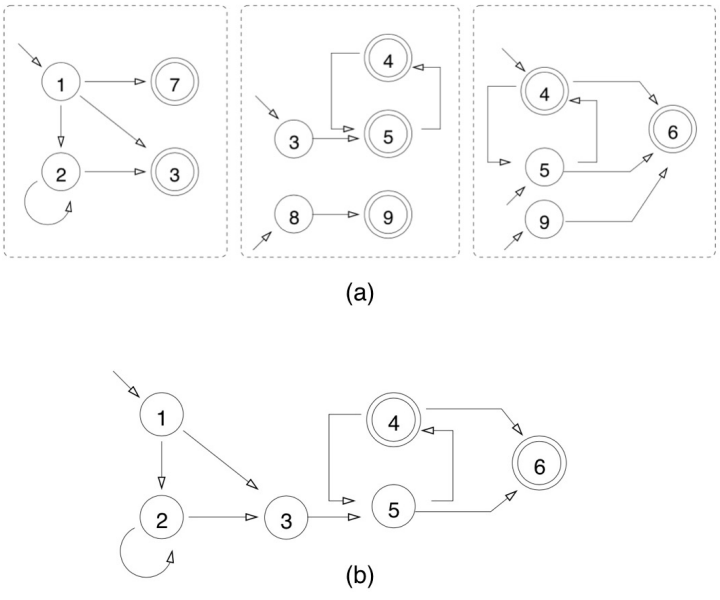


Figure 2.4 – Chaîne de trois automates (a) et la reconstruction de cet automate à partir de la chaîne (b).

Les états 7, 8 et 9 ont été supprimés par l'opération de reconstruction (dans une étape de simplification) puisqu'ils ne peuvent appartenir à aucune trajectoire. Toutes les opérations nécessaires au calcul du diagnostic adapté à une représentation par chaîne d'automates sont données en détail dans la publication [GCL05b], telles que la concaténation de chaîne, le découpage en chaîne d'automates, la reconstruction de trajectoires, la synchronisation d'une chaîne avec un automate.

2.2.2 Diagnostic incrémental

L'intérêt de la représentation à base de chaîne d'automates est le calcul du diagnostic de manière incrémentale qui autorise alors le traitement d'une longue série d'observations ou le calcul en ligne du diagnostic.
Considérant que MOD soit toujours le modèle de comportement du système et OBS l'automate des observations tel que $\mathcal{E}_{OBS} = (OBS^1, \cdots, OBS^n)$ est le découpage sous la forme d'une chaîne d'automates, le *diagnostic par morceaux* est une chaîne d'automates $\mathcal{E}_\Delta = (\Delta_1 \cdots \Delta_n)$ tel que $\mathcal{E}_\Delta$ est définie comme suit : $\mathcal{E}_\Delta = \mathcal{E}_{OBS} \otimes MOD$ [GCL05c].

La manipulation d'automates de taille réduite accélère les performances et permet même la parallélisation du calcul des morceaux de diagnostic. On parle de *découpage temporel* lorsque chaque automate de la chaîne des observations est associé à une fenêtre temporelle. Cette solution est plus adéquate pour le diagnostic bien que des propriétés restent néanmoins à vérifier afin de garantir que le découpage soit correct [GCL05a]. Outre la représentation du diagnostic sous forme de chaîne, la contribution majeure de ce travail est le calcul incrémental du diagnostic permettant de déterminer l'état final du système sans avoir à reconstruire le diagnostic complet. Grâce à une nouvelle définition de l'opération de synchronisation, dite incrémentale [GCL05a], le calcul du diagnostic par étape peut être réalisé selon deux boucles, de type *prévision-postdiction* [LC00]. Lorsqu'il s'agit de superviser le comportement d'un système, le calcul du diagnostic doit être réalisé *en-ligne* et cette contribution permet de pallier aux difficultés précédemment mentionnées concernant la complexité et l'efficacité des calculs. La méthode a été étendue au cas du diagnostic en ligne en étudiant trois types de fenêtres temporelles : les *fenêtres sûres*, les *fenêtres non sûres* et les *fenêtres tampons*. Une fenêtre temporelle est sûre si elle est délimitée par des points d'arrêt sains, c'est-à-dire qu'aucune observation émise avant ce point d'arrêt ne peut être reçue plus tard. Si ce cas de figure à l'avantage de faciliter le calcul, la détermination des points d'arrêt n'est pas toujours évidente, surtout si le flot des observations est dense et constant. Les découpages temporels en fenêtres non sûres et tampons s'appuient sur le modèle pour prédire les observations qui pourraient être reçues en fixant un délai plus ou moins long. Cette méthode présente l'inconvénient de faire croître la

chaîne d'automates et rend alors plus complexe le calcul incrémental du diagnostic (les définitions sont données en détail dans la thèse d'Alban Grastien [Gra05]). En pratique, seules les fenêtres sûres peuvent s'appliquer aux systèmes qui nous intéressent. Le principe consiste à établir un diagnostic Δ_{i-1} pour chaque date t_{i-1} délimitant la fenêtre temporelle $\mathcal{W}^{i-1}$ puis à utiliser ce diagnostic établi pour calculer le nouveau diagnostic Δ_i de la fenêtre temporelle $\mathcal{W}^i$. Le calcul du diagnostic incrémental est donc défini de la manière suivante.

Diagnostic Incrémental
- $\Delta_1 = (MOD \otimes OBS^1)$, et
- $\forall i \neq 1, \Delta^i = (MOD^- \otimes OBS^i)[F_\Delta^{i-1}]$ avec :
  - MOD^- définissant les états de MOD comme étant tous initiaux
  - F_Δ^{i-1} représentant l'ensemble des états finaux de Δ^{i-1}.

L'opération de restriction $[F_\Delta^{i-1}]$ supprime de l'ensemble des états initiaux de l'automate issu de la synchronisation tous les états n'appartenant pas à F_Δ^{i-1}. L'objectif étant de calculer une trajectoire couvrant la totalité de la fenêtre d'observation, chaque diagnostic partiel a comme ensemble des états initiaux, l'ensemble des états finaux du diagnostic de la fenêtre précédente. On peut retrouver plus de détails sur ces travaux dans la publication suivante [GCL05c].

2.2.3 Diagnostic décentralisé

Le cas du diagnostic décentralisé s'intéresse aux systèmes constitués d'un ensemble de composants interconnectés, la complexité du modèle globale étant inhérente aux interactions partagées entre les composants. L'approche présentée plus haut a été adaptée afin de calculer un diagnostic par fenêtre pour chaque composant. Le diagnostic d'un composant $\mathcal{C}$ pour la fenêtre temporelle $\mathcal{W}^i$ peut donc se définir de la façon suivante

Diagnostic d'un composant $\mathcal{C}$ dans la fenêtre temporelle $\mathcal{W}^i$
$\Delta_\mathcal{C}^i = (MOD_\mathcal{C}^- \otimes OBS^i)[F_\Delta^{i-1} \downarrow \mathcal{C}]$
avec $[F_\Delta^{i-1} \downarrow \mathcal{C}]$ la projection sur les états finaux de Δ_{i-1} du composant $\mathcal{C}$.

L'algorithme proposé par Pencole, Cordier et Rozé [PCR02] repose sur une propriété d'indépendance des états des sous-systèmes permettant une fusion efficace des diagnostics. Le découpage par fenêtre temporelle repose pour sa part sur une propriété d'indépendance des transitions entre les sous-systèmes. Dans cette configuration, il est difficile de garantir l'indépendance des états, des liens pouvant exister entre les états initiaux des composants et potentiellement perdus lors de la projection sur chaque composant. Une solution reposant sur un calcul efficace, sans recours à la synchronisation, d'un *diagnostic abstrait* et garantissant la

présence de tous les états finaux dans les diagnostics locaux [GC06] a été proposée. Si les résultats montrent sur des cas d'école, diagnostic hors-ligne et observations ordonnées, de meilleures performances dans l'usage de cette technique, des difficultés se posent néanmoins pour le diagnostic de systèmes réels. La principale concerne le découpage en fenêtres temporelles, la sélection d'une bonne fenêtre temporelle étant fondamentale pour garantir le résultat.

2.2.4 Diagnostic de systèmes reconfigurables

La plupart des méthodes de diagnostic reposent sur un modèle de comportement stable. Un système est reconfigurable s'il modifie sa structure au cours du temps, en ajoutant ou en supprimant des composants. Cette opération de reconfiguration peut s'appliquer dans les réseaux de télécommunication ou d'électricité, par exemple, pour lesquels on effectue des mises à jour avec l'ajout de nouveaux composants ou lorsque des mesures de prévention s'imposent en cas de panne, d'attaques, et où il est alors nécessaire d'isoler le composant incriminé (suppression de composant dans le modèle). Un modèle stable du système est appelé une *topologie* et une modification de cette topologie est appelée une *reconfiguration*. L'ajout d'une nouvelle connexion au sein du système est appelée une *action de connexion* alors que la suppression d'une connexion, une *action de déconnexion*. Chacune de ces actions est considérée comme instantanée, ce qui signifie qu'aucun événement ne peut se produire lorsque la topologie du système change. Une reconfiguration est alors définie comme un ensemble d'actions de connexion et de déconnexion. Une reconfiguration ne peut pas avoir lieu sur un état quelconque du système. Par exemple, il parait compliqué de déconnecter une pompe lorsque celle-ci délivre de l'eau. La représentation du modèle telle que définie précédemment lors de nos précédents travaux a donc été étendue afin d'y adjoindre la notion de *point de configuration* associée à un état du système autorisant une reconfiguration. Les définitions sont proposées plus formellement dans l'article suivant [GCL04], ainsi que la propriété de *configuration sûre* garantissant que les états du système restent inchangés par la reconfiguration. L'approche du diagnostic décentralisé a été étendu dans ce travail au cas des systèmes reconfigurables. Les actions de reconfiguration sur le système sont effectuées par un opérateur qui en informe le système de diagnostic. Le découpage en fenêtres temporelles est donc ici guidé par les dates de reconfiguration. Des reconfigurations plus fictives peuvent également être représentées afin de profiter des avantages du découpage en chaînes d'automates. Ces différentes méthodes de calcul du diagnostic sur des systèmes potentiellement reconfigurables, à l'aide de chaînes d'automates, de fenêtres temporelles ont été implémentées dans l'outil RAG_e par Alban Grastien durant sa thèse [Gra05].

Pour conclure sur cette partie, les contributions apportées par ce travail ont pro-

posé des solutions justes et complètes, bien que parfois complexes, pour résoudre le problème difficile du diagnostic incrémental et décentralisé. L'approche en-ligne avec des observations imprécises reste encore un problème de recherche. Zhao [ZD08] a depuis ce travail proposé une méthode permettant le diagnostic reposant sur deux fenêtres temporelles successives évitant la recherche des points d'arrêts.

2.3 Autres méthodes de diagnostic

Si les précédentes méthodes reposent sur des principes algorithmiques classiques laissant une large liberté dans les traitements, nous avons vu qu'elles nécessitent la mise en œuvre de nombreuses techniques pour pallier la complexité. Cette section aborde d'autres approches qui considèrent également le problème du diagnostic comme une recherche de chemins dans un modèle. J'ai abordé deux façons de reconsidérer ce problème. La première à l'aide des méthodes formelles proposant une représentation plus compacte du modèle et des algorithmes efficaces, souvent résumés dans le terme *model-checking*. Le model-checking partage comme le diagnostic la représentation du modèle du système à l'aide de SED ce qui facilite le pont entre ces deux domaines. La seconde méthode s'appuie sur un solveur de constraintes en Answer Set Programminng (ASP).

2.3.1 Diagnostic par model-checking

Dans ce travail, je me suis intéressée à la problématique du diagnostic *en-ligne* de *systèmes dynamiques* pour lesquels l'information de temps *datée* est essentielle. Il me semblait intéressant d'utiliser les techniques de model-checking, dédiées habituellement à la vérification de propriétés des systèmes temps réel, au calcul du diagnostic. Le problème du diagnostic a donc été formalisé de la manière suivante :

— déterminer les trajectoires qui expliquent les observations,
— extraire le diagnostic à partir de ces observations.

Je détaille particulièrement ce travail dans [CL01] dont les principes ont été les précurseurs de ceux mis en œuvre dans le chapitre suivant. Avant d'aborder la contribution au diagnostic, je présente les principes du model-checking.

model-checking symbolique
Issu du domaine des méthodes formelles, le *model-checking* est utilisé pour la vérification automatique de systèmes complexes [CGP02, BBF$^+$01]. Les propriétés analysées par model-checking sont de nature qualitative, comme par exemple : "*Est-il possible que l'ascenseur voyage la porte ouverte ?*".

Le modèle d'un système est décrit sous la forme d'un système à événements discrets (automates ou réseaux de Petri le plus souvent) et les propriétés sont exprimées à l'aide d'une logique temporelle. Le problème du *model-checking* peut alors se résumer ainsi : étant donné M un modèle du système et φ une propriété à vérifier, est-ce que M satisfait φ ? La réponse apportée par le *model-checker* est une réponse logique (vrai ou faux) sur la satisfiabilité de la formule par le modèle (cf. figure 2.5).

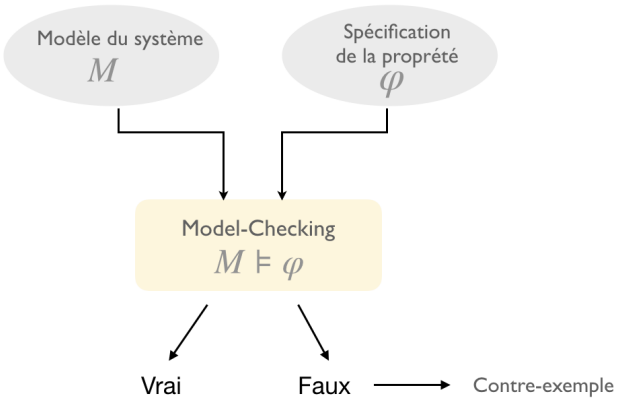


FIGURE 2.5 – Principe du model-checking.

On parle de *model-checking symbolique* [HNSY94] lorsque les méthodes ne cherchent plus à représenter de manière explicite tous les états et transitions du modèle mais utilisent des techniques symboliques pour représenter des ensembles très grands d'états dans le but de les manipuler par paquets. Ces ensembles d'états peuvent être représentés par des prédicats sur les variables d'états tel qu'un état appartient à un ensemble si et seulement si le prédicat est satisfait par cet état. Les variables d'états peuvent également être représentées de manière booléenne et le prédicat est alors une fonction booléenne représentée par des diagrammes de décision binaires (ou BDD pour Binary Decision Diagrams). Un grand nombre de model-checkers symboliques basés sur les BDD ont été employés pour la vérification de systèmes industriels.

Les propriétés à vérifier sur un modèle peuvent être classées en plusieurs familles [BBF+01] :

— *l'atteignabilité* : une certaine situation peut être atteinte,
— *l'invariance* : tous les états du système satisfont une même propriété,
— *la sûreté* : quelque chose (qu'on aimerait éviter) ne se produit jamais,

— *la vivacité* : quelque chose finira par arriver,
— *l'équité* : quelque chose se répète infiniment.

Le *model-checking* repose sur un formalisme de représentation du modèle et une logique de représentation des propriétés à vérifier. Je me suis intéressée aux systèmes dynamiques pour lesquels on s'intéresse de manière quantitative au temps, c'est-à-dire que les événements sont datés ainsi que les conditions de leur occurrence sur les états ou transitions. Le modèle le plus approprié est celui des *automates temporisés* pour lesquels la logique de description des propriétés est généralement TCTL (Timed Computation Tree Logic). Les algorithmes de model-checking portant sur les automates temporisés sont complexes, un automate temporisé ayant la particularité de posséder un nombre infini de comportements [ACD93]. La recherche dans ce domaine a progressé avec, en particulier, les travaux de l'université d'Uppsala qui propose un outil largement répandu pour le model-checking des automates temporisés appelé UPPAAL [LPY97].

Automate Temporisé (Timed Automata)

Les automates temporisés permettent de modéliser de façon abstraite des systèmes munis de contraintes temporelles à l'aide de variables appelées *horloges*. Leur formalisme étend celui des automates avec un ensemble de variables d'horloge dont les valeurs croissent uniformément avec le passage du temps. Les sommets de l'automate représentent les états et les arcs sont étiquetés par un label qui correspond à un événement ou à une action. Toutes les variables associées à des horloges physiques avancent de manière synchrone. Elles sont utilisées pour exprimer le temps de séjour dans un état de l'automate ou pour conditionner le passage d'un état vers un nouvel état. Un automate peut comporter plusieurs horloges indépendantes entre elles. Les transitions sont toujours instantanées et autorisent une remise à zéro des horloges. La contrainte d'horloge associée à un état est appelée l'*invariant* et indique que le système peut rester dans cet état tant que la condition représentée par l'invariant reste vérifiée. La contrainte d'horloge associée à une transition est appelée *garde* et conditionne le passage d'un sommet à un autre sommet de l'automate.

Contraintes temporelles.
Les contraintes temporelles définies par Alur [AD90, AD94] portent sur les horloges et conditionnent l'exécution des transitions ou le temps de séjour dans les sommets. Une contrainte d'horloge est définie comme la conjonction de contraintes atomiques. Une contrainte atomique compare la valeur de l'horloge avec un nombre entier. Si $\mathcal{X}$ est l'ensemble des variables d'horloge, on définit par $\Phi(\mathcal{X})$, l'ensemble

des contraintes temporelles possibles φ portant sur les horloges.

Contrainte temporelle
$\varphi ::= x \leq c \mid c \leq x \mid x < c \mid c < x \mid \varphi \wedge \varphi$ avec :
— $x \in \mathcal{X}$ une horloge
— c une constante telle que $c \in \mathbb{Q}$

Interprétation d'horloge
Une interprétation d'horloge v est une fonction qui associe une valeur réelle positive à chaque horloge $x \in \mathcal{X}$ telle que $v(x) \in R^+$.

Syntaxe et sémantique.
La figure 2.6 représente un automate temporisé constitué de deux sommets *Open* et *Closed* et d'une horloge x. Le système peut rester à son état initial *Open* tant que la valeur de l'horloge est inférieure à 10 ce qui est défini par l'invariant $x \leq 10$ associé à ce sommet. À partir d'une valeur de 8, définie par la garde $x \geq 8$, le système peut transiter vers *Closed* par la transition étiquetée par $a1$. Dès que l'horloge atteint la valeur de 15, la transition vers *Open* est possible grâce à la garde $x \geq 15$ sur la transition étiquetée par $a2$. L'horloge x est remise à zero lorsque la transition est tirée et le système revient à l'état *Open* avec une horloge égale à 0.

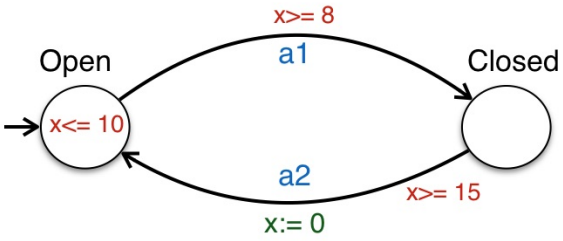


Figure 2.6 – Exemple d'automate temporisé. Les contraintes temporelles, invariant et gardes, sont représentées en rouge.

Le modèle des automates temporisés a été initialement défini par Alur et Dill [AD90, AD94] mais a pu souvent être étendu dans des variantes de la définition suivante.

Automate Temporisé (Alur & Dill [AD90, AD94])
Un automate temporisé A est défini par un n-uplet $< \mathcal{S}, \mathcal{X}, \mathcal{L}, \mathcal{E}, \mathcal{I} >$ avec :

- $\mathcal{S}$ est un ensemble fini de sommets, $s_o \in S$ étant le sommet initial ;
- $\mathcal{X}$ est un ensemble fini d'horloges ;
- $\mathcal{L}$ est un ensemble fini d'étiquettes ;
- $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{L} \times \Phi(\mathcal{X}) \times 2^{\mathcal{X}} \times \mathcal{S}$ est un ensemble fini d'arcs ; chaque arc e est un n-uplet $(s, l, \varphi, \delta, s')$ tel que e relie $s \in \mathcal{S}$, sommet source, à $s' \in \mathcal{S}$, sommet destination sur une étiquette l ; la condition d'activation que doit satisfaire les horloges pour franchir la transition est représentée par φ et l'ensemble $\delta \subseteq \mathcal{X}$ correspond à l'ensemble des horloges réinitialisées lorsque la transition est tirée ;
- $\mathcal{I} : S \to \Phi(\mathcal{X})$ associe à chaque sommet de l'automate temporisé une contrainte temporelle φ appelée l'*invariant* du sommet.

Le comportement de l'automate est le suivant. Un *état* de A est une paire (s, v) telle que s est un sommet de $\mathcal{S}$ et v une interprétation pour $\mathcal{X}$ telle que v satisfait l'invariant $\mathcal{I}(s)$. À l'initialisation, le système se trouve dans l'état initial (s_0, v_0) où s_0 est le sommet initial et v_0 est une interprétation telle que pour tout x, $v_0(x) = 0$. Toutes les transitions sont instantanées. Les valeurs des horloges croissent uniformément avec le passage du temps. À tout instant, la valeur d'une horloge est égale au temps écoulé depuis la dernière fois où elle a été mise à zéro. Le système peut séjourner dans un sommet tant que l'invariant $I(s)$ est vérifié par les valeurs d'horloges.

La sémantique d'un automate temporisé est donnée par un système de transitions étiquetées. $< \mathcal{Q}, \to, (s_o, v_o) >$ où $\mathcal{Q}$ est l'ensemble des états, $\to$ est l'ensemble des transitions étiquetées et (s_o, v_o) est l'état initial. Chaque transition représente le fait de franchir un arc ou simplement de laisser passer le temps dans un sommet. Les deux types de transition sont donc les suivantes :

Transition

- *Transition discrète.* Soit $e = (s, l, \varphi, \delta, s') \in \mathcal{E}$ une transition. L'état (s, v) a une transition discrète vers l'état (s', v') si v satisfait φ. Cette transition est notée : $(s, v) \xrightarrow{l} (s', v')$ avec $v' = v[\delta := 0]$.
- *Transition temporelle* Soit $t \in R^+$. L'état (s, v) a une transition temporelle vers l'état $(s, v + t)$ notée $(s, v) \xrightarrow{t} (s, v + t)$ si pour tout $t' \leq t$, $v + t'$ satisfait l'invariant $\mathcal{I}(s)$. La valuation $v + t$ est la valuation obtenue en augmentant toutes les horloges de t.

La théorie des automates temporisés permet de décrire un système complexe comme un produit d'automates. Le comportement du système global est obtenu

par la composition parallèle de plusieurs automates se synchronisant sur les étiquettes des arcs. Lors de la définition des composants, il est alors nécessaire de distinguer les événements internes décrivant la dynamique propre et asynchrone du sous-système, des événements de synchronisation qui provoquent l'évolution simultanée de tous les sous-systèmes partageant ces événements.

PRODUIT D'AUTOMATES TEMPORISÉS

Le produit de deux automates :
$\mathcal{A}_1 =< \mathcal{S}_1, \mathcal{X}_1, \mathcal{L}_1, \mathcal{E}_1, \mathcal{I}_1 >$ et $\mathcal{A}_2 =< \mathcal{S}_2, \mathcal{X}_2, \mathcal{L}_2, \mathcal{E}_2, \mathcal{I}_2 >$ noté $\mathcal{A}_1 \,||\, \mathcal{A}_2$ est un automate temporisé $\mathcal{A}$ tel que $\mathcal{A} =< \mathcal{S}_1 \cup \mathcal{S}_2, \mathcal{X}_1 \cup \mathcal{X}_2, \mathcal{L}_1 \cup \mathcal{L}_2, \mathcal{E}, \mathcal{I} >$ où :

- $\forall (s_1, s_2) \in \mathcal{S}_1 \times \mathcal{S}_2$, $\mathcal{I}(s_1, s_2) = \mathcal{I}_1(s_1) \wedge \mathcal{I}_2(s_2)$
- $\mathcal{E}$ est un ensemble de transitions construit de la façon suivante :
  - pour tout $l \in \mathcal{L}_1 \cap \mathcal{L}_2$, pour tout $(s_1, l_1, \varphi_1, \delta_1, s_1') \in \mathcal{E}_1$ et $(s_2, l_2, \varphi_2, \delta_2, s_2') \in \mathcal{E}_2$, $((s_1, s_2), l, \varphi_1 \cup \varphi_2, \delta_1 \wedge \delta_2, (s_1', s_2')) \in \mathcal{E}$
  - pour tout $l \in \mathcal{L}_1 \setminus \mathcal{L}_2$, pour tout $(s_1, l_1, \varphi_1, \delta_1, s_1') \in \mathcal{E}_1$ et $s_2 \in \mathcal{E}_2$, $((s_1, s_2), l, \varphi_1, \delta_1, (s_1', s_2')) \in \mathcal{E}$
  - pour tout $l \in \mathcal{L}_2 \cap \mathcal{L}_1$, pour tout $s_1 \in \mathcal{E}_1$ et $(s_2, l_2, \varphi_2, \delta_2, s_2') \in \mathcal{E}_2$, $((s_1, s_2), l, \varphi_2, \delta_2, (s_1', s_2')) \in \mathcal{E}$

Logique TCTL (*Timed Computation Tree Logic*)

La logique la plus répandue pour la vérification d'automates temporisés est la logique TCTL (Timed Computation Tree Logic) [HNSY94]qui est une extension de la logique CTL autorisant l'expression de contraintes temporelles dans les propriétés. La logique CTL est interprétée sur des structures de Kripke et permet d'exprimer des propriétés sur ces états en faisant intervenir l'arbre d'exécutions issu de ces états. Le principe adopté par la logique TCTL pour introduire la dimension temporelle consiste à borner la portée des opérateurs dans le temps et à introduire des contraintes d'horloge dans les formules. Les formules TCTL sont définies de manière inductive par la grammaire suivante :

LOGIQUE TCTL

$$f ::= p \mid x \in \mathcal{I} \;\mid\; \neg f \;\mid\; f_1 \vee f_2 \;\mid\; \exists \Diamond_I f \;\mid \forall \Diamond_I f \;\mid\; \exists \Box_I f \;\mid\; \forall \Box_I f$$

avec $p \in P$ un prédicat de base, $x \in \mathcal{X}$ une horloge et I un intervalle de temps.

On interprète les formules sur un système de transitions à l'aide de deux quantifieurs de chemin, $\exists$ et $\forall$, et de deux opérateurs temporels, $\Diamond$ et $\Box$, qui expriment la sémantique de la formule sur une trace.

- $\forall$ (noté aussi **A** pour *Always*) : la formule est vraie si elle est vérifiée pour tous les successeurs

— $\exists$ (noté aussi **E** pour *Exists*) : la formule est vraie si elle est vérifiée pour au moins un successeur
— $\Diamond$ (noté aussi **F** pour *Finally*) : la formule est vraie si elle est vérifiée dans un état futur du chemin
— $\Box$ (noté aussi **G** pour *Globally*) : la formule est vraie si elle est vérifiée dans tous les états suivants du chemin ;

Intuitivement $\exists\Diamond_I f$ signifie qu'il existe une exécution menant à une configuration satisfaisant la propriété f au temps $t \in I$. $\forall\Diamond_I f$ signifie que toutes les exécutions possèdent une configuration où la propriété f est valide au temps $t \in I$. $\exists\Box_I f$ indique qu'il existe une exécution telle que toutes les configurations sur le chemin de l'exécution satisfont la propriété f. $\forall\Box_I$ signifie que, pour toutes les exécutions, toutes les configurations satisfont la propriété f.

Calcul des trajectoires par model-checking
Dans ce travail, j'ai proposé une méthode de calcul du diagnostic par model-checking sur des systèmes temporisés [CL01]. Le problème de diagnostic est généralement composé de deux parties : la description du système et le langage des observations. Ici, le système est représenté à l'aide d'un automate temporisé obtenu par composition des différents modèles de composant. Une illustration sur un système de pompe, pour lequel on cherche un composant défectueux, est détaillé ci-dessous. On considère un système de transitions basé sur les états (structure de Kripke), ce qui signifie que les observables, fournis par les capteurs des composants, sont un sous-ensemble des propriétés des états du système. Les observations sont potentiellement imprécises ce qui autorise un ensemble de propriétés associées à une observation pour une date donnée. Le système global est obtenu à l'aide de la synchronisation des composants sur les valeurs des flux en sortie des sous-systèmes, la pompe et le tuyau, au niveau des points de synchronisation x et y (cf. figure 2.7). Soit $\mathcal{A}$ le modèle global du système, et I_{s_i} la contrainte temporelle décrivant l'intervalle de temps associé à chaque état s_i et notée $\mathcal{K}_{\mathcal{A}}(I_{s_i})$, le modèle temporel est défini par la paire : $(\mathcal{A}, \mathcal{K}_{\mathcal{A}}(I_s))$.

Les observations instantanées Obs sont associées à des dates t_{Obs}. Une observation datée est une paire $(Obs, \mathcal{K}_{Obs}(t_{Obs}))$. Il est à noter que chaque observation peut être associée à un ensemble de propriétés observables du système noté O_ϕ.
Une séquence d'observations datées OBS est une paire $([Obs_1, \ldots, Obs_p], \mathcal{K}_{OBS}(t_{Obs_1}, .., t_{Obs_p}))$ où chaque Obs_i est une observation et où $\mathcal{K}_{OBS} = \bigcup_{i\in[1\ldots p]} \mathcal{K}_{Obs_i}(t_{Obs_i})$ collecte les contraintes associées à chaque Obs_i. On considère qu'une trajectoire explique une séquence d'observations datées, s'il existe un chemin au sein de l'automate permettant d'associer chaque propriété des états traversés avec chacune des valeurs observées tout en respectant les contraintes tem-

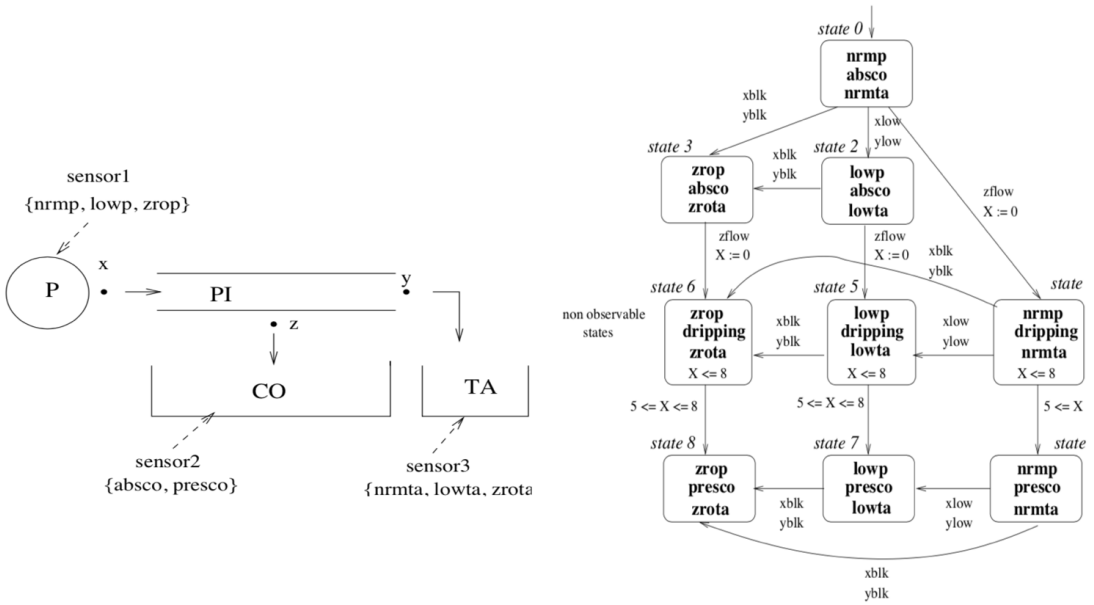


FIGURE 2.7 – Système de pompe et modèle global sous forme d'automate temporisé issu de la synchronisation des modèles des composants

porelles. Toutes les transitions observables ne nécessitent pas nécessairement d'être observées, ce qui peut être le cas si un capteur est altéré. Le notion d'"explicabilité" entre les observations et les états du système peut être interprété dans le sens "implique", cependant certaines descriptions du domaine peuvent considérer des lois statiques causales reliant les événements ou actions aux états du système [BMS00].

Une *trajectoire temporelle* $(Traj(s,s'), \mathcal{K}_{Traj}(I_s))$ *explique* une séquence de p observations datées $(([Obs_1,\ldots,Obs_p], \mathcal{K}_{OBS}(t_{Obs_1},..,t_{Obs_p}))$ si et seulement si il existe une projection sur les observables des états de la trace $Proj_s(Traj(s,s')) = [O\phi_1,\ldots,O\phi_n]$, $O\phi_i$ étant la projection observable d'un état telle que :

- il existe une fonction de correspondance *map* de $[1\ldots p]$ vers $[1\ldots n]$
- $\forall i \in [1\ldots p]$ $Explain(O\phi_{map(i)}, Obs_i)$ est satisfaite. $Explain(O\phi, Obs)$ signifie que $\exists O \in Obs$ tel que $O \subseteq O\phi$.
- Les contraintes de la séquence des observations satisfait les contraintes de la trajectoire i.e : $\{\mathcal{K}_{OBS}(t_{Obs_1},..,t_{Obs_p}) \bigcup \forall i \in [1\ldots p], t_{Obs_i} \in I_{s_{map(i)}}\} \models \mathcal{K}_{Traj}(I_s)$.

L'intérêt de la méthode consiste à traduire cette relation entre le modèle et les

observations à l'aide d'une formule TCTL qu'il est possible de vérifier par model-checking. Soit MC le model-checker utilisé, A le modèle global du système obtenu par synchronisation des modèles des composants, Obs_i l'ensemble des observations obtenues associées aux contraintes temporelles I_{Obs_i}, le calcul de la trajectoire entre deux observations peut être exprimé de la manière suivante à l'aide d'une formule TCTL : $TRAJ := MC[\mathcal{A}, s \wedge x \in I_{Obs_i} \Rightarrow \exists\Diamond_{I_{Obs_{i+1}}} s']$. Un algorithme incrémental permet de reconstituer la trajectoire à l'aide des morceaux de trajectoire vérifiés entre deux observations par le model-checker. Il repose également sur un modèle d'observations qui peut être considéré comme un modèle inversé du système global. Il est alors possible d'isoler les événements de fautes à partir de cette trajectoire. Soit $\mathcal{L}^{-1}$ le modèle d'observations, x l'horloge du modèle A, K_i l'ensemble des états possibles au temps t_i, MC un model-checker sur automates temporisés, le diagnostic sur n observations est calculé par l'Algorithme 1.

Algorithm 1 Diagnostic sur n observations :

Initialisation :
$K_1 := \emptyset$
for all $O_{1j} \in Obs_1$ **do**
  $K_1 := K_1 \cup \mathcal{L}^{-1}(O_{1j})$
end for
for all pairs (Obs_i, Obs_{i+1}) with $i \in [1, n-1]$ **do**
  $S_{(i+1)} := \emptyset$
  for all $O_{(i+1)j} \in Obs_{i+1}$ **do**
    $S_{(i+1)} := S_{(i+1)} \cup \mathcal{L}^{-1}(O_{(i+1)j})$
  end for
  for all $s \in K_i$ and pour tout $s' \in S_{(i+1)}$ **do**
    $TRAJ' := MC[\mathcal{A}, s \wedge x \in I_{Obs_i} \Rightarrow \exists\Diamond_{I_{Obs_{i+1}}} s']$
    $K_{i+1} := K_{i+1} \cup s'$
    $TRAJ := Build(TRAJ', s, s')$
  end for
end for

Cette méthode peut être particulièrement intéressante dans le cadre du diagnostic en ligne qui permet de calculer la trajectoire du système au fur et à mesure des observations en tirant partie de l'efficacité des model-checkers.

2.3.2 Diagnostic avec ASP

Si le model-checking est une première voie pour réduire la complexité, une méthode duale repose sur une représentation logique avec l'usage des techniques de satisfiabilité propositionnelle (SAT) [RG07] [GA08]. Ce travail a été réalisé dans le cadre du stage de Master 1 de François Laferrière que j'ai co-encadré avec Laurence Rozé.

La technique suit le même principe qu'avec le model-checking : la recherche de chemin est transformée en une formule propositionnelle satisfiable si et seulement si un chemin existe. L'occasion s'est présentée d'évaluer l'efficacité du paradigme Answer Set Programming (ASP) par rapport à celui de SAT pour la recherche du diagnostic : Torsten Schaub chercheur en logique de l'université de Postdam a été accueilli, dans le cadre d'une chaire INRIA, dans l'équipe Lacodam de 2014 à 2018. Il a développé un solveur ASP appelé *clingo* [GKK+11].

Cette étude s'appuie sur un travail d'Alban Grastien (aujourd'hui chercheur à l'université de Canberra en Australie) qui propose une méthode de diagnostic à l'aide de SAT [GA13]. Answer Set Programming est une forme de programmation déclarative par contraintes basée sur une logique non monotone utilisée pour la représentation des connaissances, le raisonnement et la résolution de problèmes déclaratifs. La modélisation du système qui correspond au fonctionnement d'un ensemble d'automates communicants peut s'exprimer par des prédicats logiques qui permettent de représenter des composants, les connexions entre ces composants et les règles de transitions (état de départ, événement déclencheur, état terminal, événements générés).

Prenons le composant (1,1) qui passe de l'état normal (n) à (f) s'il reçoit un événement de faute de la part du composant (1,0) et déclenche alors un *reboot*. La représentation en ASP de cette transition est la suivante :

```
edge((comp(1,1),2), state(comp(1,1), n), event(comp(1,0), fault),
                    event(comp(1,1), reb), state(comp(1,1), f)).
```

Nous avons considéré trois types d'observations : les observations datées (la date de l'observation est connue précisément), les observations partiellement ordonnées (l'ordre des observations est connu mais pas leur date d'occurrence exacte), les observations partiellement ordonnées (l'ordre total n'est pas connu, pour chaque observation seules quelques observations la précédant sont indiquées) (cf. table 2.1 pour leur représentation en ASP).

Contrairement aux automates temporisés qui gèrent de manière efficace et transparentes les horloges, nous sommes ici contraints de simuler le pas de temps

`obs(event(comp(0,1),back),6)`	Observation datée Le second argument définit le pas de temps

`obs(event(comp(0,1),reb),1)`
`obs(event(comp(2,1),reb),2)`
`obs(event(comp(0,1),back),3)`
	Observations ordonnées
	Le second argument définit l'ordre

`obs(event(comp(2,1),back),1)`
`precede(1,4)`
`obs(event(comp(0,0),reb),2)`
`obs(event(comp(0,3),reb),3)`
`precede(3,2)`
`precede(3,4)`
`obs(event(comp(2,1),reb),4)`
	Observations partiellement ordonnées
	Le second argument définit un indice
	Nécessité d'un prédicat *precede*
	Définit un ordre entre deux observations

TABLE 2.1 – Représentation des trois types d'observations en ASP

à l'aide de prédicats. Le diagnostic s'effectue alors en quatre étapes : l'initialisation, le choix des événements, la propagation des événements et l'inertie (qui fait évoluer le temps au sein d'un composant dont l'état reste inchangé) et enfin la vérification de la satisfaction des contraintes. Trois algorithmes différents ont été nécessaires afin de traiter les différents types d'observation. Si les résultats sur les observations datées sont bien meilleurs que pour la solution SAT, le problème de la gestion temporelle simulée complexifie de manière exponentielle les calculs pour les deux autres types de jeux d'observations [Laf16]. Générer l'ensemble des chemins possibles est extrêmement rapide en ASP mais les éliminer s'ils ne correspondent pas à l'ordre des observations est une opération plus complexe difficilement soluble. ASP est un langage de haut-niveau qui permet de définir aisément le problème sous une forme logique, cependant notre expérience, comme d'autres, montrent ses limites pour représenter et raisonner sur des systèmes dynamiques. Des travaux très récents de l'équipe de Torsten Schaub proposent une extension temporelle d'ASP implémentée dans un solveur *Telingo* [CKMS19] offrant des opérateurs temporels qui pourraient résoudre notre problème d'expression des observations partiellement ordonnées et ouvre donc des perspectives à ce travail.

Bilan sur le diagnostic par model-checking et solveur logique
La principale motivation de ces travaux étaient de tirer partie de la représentation symbolique associée à des algorithmes efficaces pour pallier au problème de combinatoire du diagnostic. Nos travaux sur le diagnostic à l'aide de model-checking sont aujourd'hui présentés comme faisant partie des précurseurs du diagnostic par model-checking [BM15]. L'usage des solveurs logiques comme SAT ou ASP, plus récents, permet un cadre unifié et une déclarativité intéressante pour les inter-

actions complexes entre états ou composants. La représentation temporelle reste cependant encore difficile à gérer pour la représentation des systèmes dynamiques, alourdit les déclarations, bride la résolution et donc l'efficacité des solveurs.
Récemment, la diagnosticabilité par model-checking [Gra09] a fait l'objet d'études plus nombreuses que le diagnostic lui-même, variant les techniques de vérification et les logiques temporelles. Généralement la définition de la diagnosticabilité à été simplifiée par rapport à celle donnée par Sampath afin de la réduire à une propriété d'atteignabilité sur un modèle synchronisé avec lui-même par la technique du *twin plant* [Gra09]. Plus récemment des travaux ont été menés sur des réseaux de Petri [BPDT17], les auteurs s'intéressant à la *k-diagnosticabilité*. Tous ces travaux partagent le même problème : la diagnosticabilité d'une seule faute sur le système. Des auteurs ont étendu la notion de faute à celle de *pattern*. Le pattern est défini par une spécification en logique temporelle sur un automate, et la panne comme un comportement qui n'a pas de correspondance avec le pattern. Gougam et al. [GPS17] propose une méthode permettant de vérifier la diagnosticabilité à l'aide de pattern décrivant des comportements d'intérêt, normaux ou anormaux à l'aide de réseaux de Petri puis d'utiliser des méthodes de model-checking pour tester un *système-pattern*.

Les techniques de model-checking sont efficaces lorsque les méthodes énumératives atteignent leur limite mais leur réponse binaire n'est pas toujours suffisante en terme d'interprétabilité et d'explication lorsque l'on se place dans un contexte d'aide à la décision. Dans la contribution sur le diagnostic, nous les avons englobé dans un algorithme qui traite au fil de l'eau les observations et construit les trajectoires du système expliquant les observations. Les perspectives de ce travail seraient d'étendre le principe à des systèmes plus complexes en particulier en levant l'hypothèse considérant que toutes les transitions sont observables ou en considérant les situations de pannes multiples. Ces idées restent des sujets d'intérêt dans le cadre des méthodes généralistes du diagnostic, mais les circonstances m'ont amenées à adapter ces approches en partant des applications, dans le domaine de l'agro-écologie.

Chapitre 3

Model-checking pour la gestion des écosystèmes

Sommaire

3.1 Enjeu 60
3.2 Modélisation qualitative des écosystèmes 60
3.3 Problématique 62
3.4 Framework de modélisation 63
3.4.1 Modélisation par automates temporisés 63
3.4.2 Génération automatique du modèle d'automates 66
3.4.3 L'écosystème marin du lagon d'Ouvéa 69
3.5 Requêtes prédictives *what-if ?* par model-checking . . . 72
3.5.1 Patrons de requêtes 72
3.5.2 Logiciel EcoMata 75
3.5.3 Expérimentations 76
3.5.4 Bilan sur l'exploration de modèles par des scénarios prédictifs 77
3.6 Requêtes proactives *what-for ?* par synthèse de contrôleur 79
3.6.1 Automate temporisé de jeu et à coûts 79
3.6.2 Synthèse de contrôleur 81
3.6.3 Planification de stratégies 83
3.7 Intégration de facteurs socio-économiques 85
3.8 Conclusion 88

3.1 Enjeu

Avec l'émergence du *développement durable*, le soucis de réduire la pression sur les ressources environnementales a nécessité la prise en considération des activités humaines sur les systèmes biologiques. L'aide à la décision dédiée auparavant aux systèmes physiques, industriels ou processus d'entreprise, s'est donc tout naturellement trouvé un nouveau champ d'application, les systèmes naturels (écosystèmes, exploitations agricoles, territoires, etc.). Les principes évoqués plus haut se retrouvent être les mêmes : prédire, expliquer, évaluer l'impact d'une décision, aider au choix des décisions. Cependant, les systèmes naturels sont des systèmes complexes qui comportent des caractéristiques très spécifiques :

- Ils sont constitués d'un grand nombre d'éléments de différentes natures (les organismes biologiques, l'environnement, les activités humaines, etc.).
- Ces éléments sont en interaction permanente selon des relations pas toujours bien connues.
- Le comportement du système est fortement non-linéaire.
- Le système est difficile à délimiter. La granularité temporelle et spatiale doit être élargie pour intégrer tous les éléments nécessaires mais suffisamment précise pour répondre au besoin spécifique des gestionnaires.
- Les données disponibles sont hétérogènes (numériques issues des capteurs ou de modèles mécanistes très précis, symboliques pour les connaissances expertes), incomplètes (certains éléments, relations ou comportements représentés dans l'écosystème ne sont pas encore bien connus), imprécises (données bruitées issues des capteurs) et incertaines (les informations peuvent être approximatives).

Les objectifs de ce travail consistent donc à proposer une modélisation permettant de mieux répondre aux questions suivantes :

- Comprendre des systèmes complexes en décrivant les interactions entre les composants biologiques et les activités humaines en tenant compte particulièrement de la composante temporelle.
- Aider les gestionnaires à superviser le système et à définir les décisions les plus adaptées pour les usagers et les ressources.

3.2 Modélisation qualitative des écosystèmes

Les modélisations utilisées par nos collègues thématiciens, écologues ou agronomes, s'appuient généralement sur des modèles mécanistes, à base d'équations mathématiques [Ryk89]. Lorsque les processus sont bien connus et que les données expérimentales précises sont disponibles, ces modèles numériques sont bien adaptés

et offrent beaucoup de précision dans les résultats. Cependant, dans de nombreux domaines, comme les systèmes marins par exemple, il peut être coûteux et difficile d'acquérir des données qui, si elles existent, peuvent être imprécises ou incomplètes. Par ailleurs, les interactions et rétro-actions entre les espèces sont parfois trop complexes pour être prise en compte dans des modèles mécanistes surtout si l'on y intègre des politiques d'exploitation. Les modèles qualitatifs s'avère être une alternative pertinente lorsque l'on s'intéresse à des systèmes complexes et mal connus [WdK89, Kui94]. Bien que souffrant d'un peu de méfiance de la part des thématiciens plus confiants dans la rigueur mathématique (mais parfois dépourvus lorsqu'il s'agit de remplir la somme des paramètres requis en entrée d'un modèle mécaniste), les modèles qualitatifs reposent pourtant sur de solides bases théoriques et proposent une abstraction fiable des systèmes étudiés [TMID03]. Les modèles qualitatifs sont souvent plus faciles à appréhender par les experts, ce qui les rend plus adaptés comme outils d'aide à la décision. Ils permettent non seulement d'améliorer la compréhension que l'on a des systèmes mais aussi de prédire les effets à long terme d'actions sur le terrain, de déterminer les valeurs critiques des variables impliquées et ainsi de mieux évaluer les actions possibles en terme de gestion et de gouvernance.
Dans le domaine de l'écologie, le raisonnement qualitatif a été réellement implémenté à partir des années 90 avec par exemple l'outil de simulation qualitative QSIM appliqué en hydroécologie [Gue91] ou pour l'étude de l'impact des zones de reproduction sur la mortalité des saumons [GD01]. Plus récemment, des modèles plus complexes ont été construits pour étudier les interactions entre les processus biochimiques, physiques, chimiques, comme [SBA06] dans le domaine des systèmes aquatiques. D'autres travaux ont étudiés plus globalement les interactions entre l'environnement, les hommes et les sous-systèmes écologiques pour analyser les effets des activités anthropiques sur les communautés de macroinvertébrés benthiques dans un bassin versant [TN06]. En écologie, la formalisation la plus populaire utilisée par les chercheurs thématiciens repose sur les QNM (Qualitative Network Analysis) aussi connue sous le terme de *loop analysis* [PL85, DHR03]. Cette représentation décrit les interactions à l'aide des signes (+,-,0) dénotant le type d'impact entre les variables du système. QNM considère que le réseau de relations du système est statique et la force de l'impact est valuée de manière fixe. Sa facilité d'usage a permis son application dans des nombreuses applications comme l'analyse des perturbations sur les communautés naturelles [DLR05], de la pêche récréative [MMG10], du déclin des saumon [SGS18] ou de l'aquaculture [RMF$^+$15]. Enfin, les approches multi-agents sont également très présentes dans le domaine, notamment avec la plateforme Cormas [BBLPB16] (le lecteur intéressé pourra se reporter à cette synthèse très détaillée sur l'usage des SMA pour la gestion de l'environnement [LPBBW13]).

3.3 Problématique

Comme nous avons pu le voir dans le chapitre précédent sur le diagnostic, même si un modèle qualitatif réduit la complexité en proposant une plus grande abstraction du système réel, son exploration reste coûteuse. Les approches précédentes s'intéressent à des systèmes où la dimension temporelle est peu présente, afin de limiter les problèmes de combinatoire permettant de garantir une interactivité acceptable pour l'outil d'aide à la décision.

Dans ce travail, l'idée consiste à vérifier qu'il est possible d'employer les méthodes de model-checking sur des automates pour la supervision et l'aide à la décision des écosystèmes. Le premier intérêt de ces techniques est de lever la contrainte du temps, non traitée par les méthodes qualitatives jusqu'à présent, en utilisant les automates temporisés que l'on sait efficaces pour la vérification.

L'objectif de cette contribution était de construire un système interactif pour l'aide à décision permettant de tester différentes stratégies de gestion des ressources et d'évaluer leurs conséquences sur l'écosystème à plus ou moins long terme. Afin d'atteindre cet objectif, plusieurs approches ont été abordées.

- Proposer une modélisation qualitative et temporelle de toutes les composantes d'un écosystème.
- Explorer l'écosystème à l'aide de scénarios de type *what-if ?* et analyser l'impact des politiques sur la dynamique de l'écosystème.
- Explorer l'écosystème à l'aide de requêtes proactives de type *what-for ?* dans un objectif de *sûreté* dans le langage informatique, de *durabilité* en écologie.
- Trouver la meilleure stratégie d'exploitation permettant d'atteindre un état souhaité pour l'écosystème à une date donnée.
- Intégrer des facteurs socio-économiques permettant d'identifier la meilleure stratégie pour l'écosystème qui soit le plus économiquement viable pour ses utilisateurs.

Nous proposons d'utiliser les automates temporisés comme formalisme de représentation du modèle qualitatif et les techniques de model-checking qui y sont associées. Les approches que nous avons développées sont les suivantes :

- Définir un framework permettant de décrire un écosystème de manière modulaire et construire le modèle global sans avoir à décrire les automates de manière exhaustive.
- Proposer des patrons de haut-niveau pour interroger le modèle selon des requêtes proactives de type *what-if ?* sans que les gestionnaires n'aient à écrire des formules logiques pour la vérification des automates par model-checking.
- Utiliser la *synthèse de contrôleur* pour proposer des stratégies répondant à

un objectif donné.
— Appliquer des techniques de planification sur des automates temporisés étendus par des coûts pour la recherche de stratégie optimale.
— Définir des stratégies économiques sous forme de règles logiques.

Ce travail que j'ai mené durant pendant plusieurs années de recherche a été réalisé en collaboration avec de nombreuses personnes comme Guy Fontenelle, professeur en halieutique en poste à l'IRD de Nouméa puis à Agrocampus Ouest et Yves-Marie Bozec docteur en écologie marine que j'ai encadré en séjour post-doctoral au sein de l'UP informatique à Agrocampus Ouest. Ces chercheurs ont apporté leurs compétences dans le domaine de l'écologie et de la gestion des pêches, complémentaires à mon approche IA et diagnostic. Des étudiants que j'ai encadrés en stage de Master 2 ont également contribué activement à ce travail : Julie Scopélitis, Yulong Zhao (qui a continué en thèse sur le sujet), Charlotte Paillette, Omar Krichen, Pierre Gaignon et Mai Dao.

3.4 Framework de modélisation

L'écosystème est représenté par trois entités en interaction $\{ES, ENV, AP\}$. Chaque entité est constituée d'un ensemble de sous-systèmes qui eux-mêmes communiquent entre eux.
— ES décrit l'écosystème naturel. Il est composé d'un ensemble d'espèces qui interagissent selon des relations de type proie-prédateur.
— ENV décrit l'environnement de l'écosystème. Il décrit les phénomènes contrôlables ou incontrôlables qui peuvent avoir un impact sur l'écosystème.
— AP décrit les pressions anthropiques. Il représente l'ensemble des forces appliquées sur l'écosystème par l'activité humaine.

La figure 3.1 illustre cette modélisation sur un écosystème marin. Les interactions qui déterminent la dynamique du système ne sont pas de même nature. Les actions anthropiques (chasse, pêche par exemple) peuvent augmenter le déclin d'une espèce alors que les actions entre espèces tendent au contraire à stabiliser la dynamique du système. C'est le cas de la relation proie-prédateur : la diminution des proies due à la prédation engendre en retour la diminution des prédateurs et conduit à la stabilisation du système à des niveaux moyens.

3.4.1 Modélisation par automates temporisés

Le principe de ce modèle générique repose sur le formalisme des automates temporisés [AD94] qui permettent à la fois la modélisation qualitative, la gestion

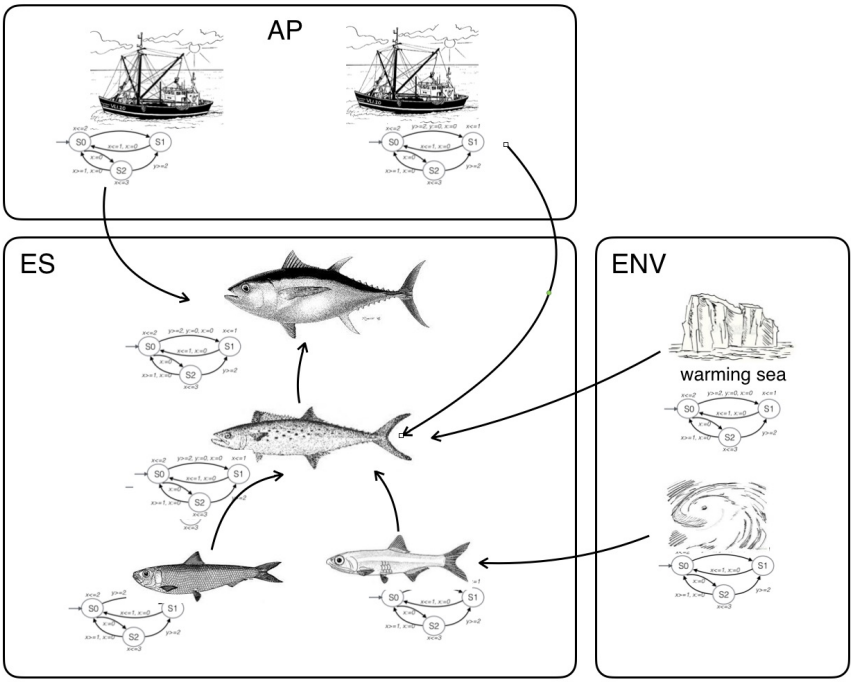


FIGURE 3.1 – Framework de modélisation d'un écosystème marin

du temps et la modularité.

Etats qualitatifs. On utilise les automates temporisés pour modéliser les sous-systèmes au sein de chaque entité. Les sommets de l'automate représentent des variables qualitatives décrivant les différents états du sous-système. Les espèces sont définies, par exemple, à l'aide de niveaux qualitatifs sur leur biomasse tels que $\{low, normal, high, extinct\}$, le domaine pouvant être défini au choix. Les forces anthropiques comme les pressions de pêche par exemple pour un écosystème marin sont également représentées par des valeurs qualitatives décrivant des pressions fortes, normales, inexistantes, etc., tout comme les phénomènes climatiques.

Interaction entre composants. L'interaction des automates temporisés est réalisée par une synchronisation de type CCS [Mil89] selon le principe décrit section 2.3.1. Deux automates se synchronisent sur une paire d'événements s'ils partagent la même étiquette. On distingue les événements émetteurs des événements récepteurs. Dans la représentation d'un écosystème, un événement peut être le déclenchement d'une activité humaine, le changement d'état d'une espèce, l'occurrence d'un catastrophe naturelle. Ces événements se situent sur des transitions qui s'exécutent

instantanément (comme le veut la définition de l'automate temporisés) dans les automates porteurs des deux événements (émetteur et récepteur). Le modèle global est obtenu par la synchronisation de tous les automates qui composent le système dans une opération appelée *composition parallèle*.

Gestion du temps. La dynamique de transition entre états peut se modéliser par les horloges des automates, chaque sous-système pouvant définir sa propre horloge. Les différences de valeurs possibles entre gardes et invariants permettent d'introduire des intervalles temporels intéressants pour représenter l'incertitude du temps de passage d'un état vers un autre état. Cet intervalle de transition est très utile lorsque l'on s'intéresse aux écosystèmes vivants en évitant un effet de seuil qui parait peu réaliste. Les transitions instantanées entre états ne posent pas de problème pour représenter l'évolution des forces anthropiques ou des événements climatiques qui surviennent rapidement. Pour les changements d'états au sein des espèces, il est nécessaire de modéliser des états intermédiaires qui décrivent la dynamique croissante ou décroissante de la biomasse entre deux états stables. Dans la modélisation, on distingue donc les *états stables* associés aux valeurs qualitatives des états du systèmes des *états intermédiaires* qui correspondent à une dynamique croissante ou décroissante du niveau qualitatif. Le système peut sortir d'un état stable uniquement par la réception d'événement (et non par le temps). Chaque état intermédiaire est associé à des contraintes temporelles qui dépendent de l'événement ayant déclenché la transition (la biomasse d'une espèce diminue moins vite si elle dispose de moins de proies qu'elle consomme modérément que si elle est exploitée intensivement, par exemple). Le nombre d'états intermédiaires peut croître rapidement selon le nombre d'interactions entre les sous-systèmes.

La figure 3.2 présente un automate temporisé de l'espèce *thon* pour un écosystème simple : thon $\leftarrow$ maquereau $\leftarrow$ {sardine, anchois} où les espèces thon et maquereau peuvent être soumis à une pression de pêche qui peut être normale, faible ou nulle. Les niveaux qualitatifs *high*, *normal* et *low* de la biomasse sont présentés. Les événements liés à la pêche au thon étiquettent les arcs entre les états stables de biomasse (en blanc) et les états intermédiaires (en noir). Lorsque l'espèce change d'état, elle déclenche un événement, *thon_high*! par exemple, qui se synchronise avec les automates voisins (la proie *maquereau* dans cet exemple). L'unité des contraintes temporelles est le mois et ces contraintes sont portées par les états intermédiaires pour indiquer la vitesse du changement. On peut ainsi remarquer, que partant de l'état *normal*, un événement indiquant l'arrêt de la pêche *peche_thon_stop*? aura pour conséquence une évolution plus rapide vers une biomasse *high* qu'une simple diminution de la pression de pêche *peche_thon_low*?. À chaque moment de l'évolution en cours, et étant donné l'échelle de temps considérée, un événement peut contredire un précédent nécessitant la remise à

jour des valeurs d'horloges, ce qui est symbolisé par les transitions entre les états intermédiaires. La description du framework de modélisation est détaillée dans la publication [LZC14].

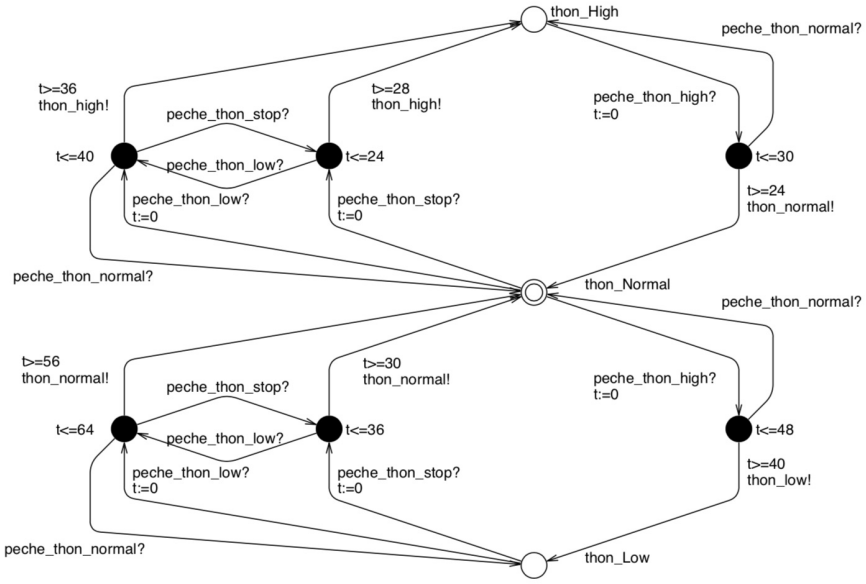


FIGURE 3.2 – Automate temporisé de l'espèce Thon soumis à une pression de pêche. La biomasse est décrite par trois niveaux qualitatifs *thon_Normal*, *thon_Low* et *thon_High*. Les états stables sont représentés en blanc et les états intermédiaires en noir.

Complexité. Soit A un réseau de n automates A_i, chacun étant défini par N niveaux qualitatifs. Pour chaque automate A_i, on définit l'ensemble des voisins $Nei(A_i) = \{A_j\}_{\{1 \leq j \leq n-1\}}$ comme l'ensemble des automates temporisés A_j tel que A_j partage au moins un événement synchronisé avec A_i. Chaque changement d'état qualitatif dans $Nei(A_i)$ peut déclencher un changement dans A_i et donc définir un état intermédiaire. Si l'on considère que chaque A_i peut avoir au plus j automates voisins, le nombre maximum d'états intermédiaires pour chaque A_i est donc N^j.

3.4.2 Génération automatique du modèle d'automates

Le modèle est rapidement trop conséquent pour être construit à la main. Nous avons donc proposé un algorithme de génération automatique du réseau d'automates à partir de données connues par les gestionnaires de l'écosystème étudié.

Les données attendues sont les suivantes.

- Pour l'entité ES, il s'agit de donner le réseau trophique des espèces et pour chaque espèce la valeur de la biomasse à l'équilibre ainsi que la productivité et le taux de consommation qui sont des informations bien connues. Les différents niveaux qualitatifs des biomasse sont définis par l'utilisateur.
- Pour l'entité AP, un chronogramme des pressions anthropiques appliquées dans le temps est fourni.
- Pour l'entité ENV, la définition des perturbations possibles, éventuellement associées à un délai (pour le réchauffement climatique par exemple) peut être énoncé.

Calcul des contraintes temporelles. Si le modèle repose sur des états qualitatifs, il est nécessaire d'acquérir les informations temporelles qui sont des données numériques (éventuellement incertaines). Pour les perturbations environnementales et les activités humaines, ces données sont issues du gestionnaire. Pour l'évolution des états de la biomasse des espèces soumises aux diverses pressions, ces informations doivent être calculées. Nous nous sommes reposés sur le modèle de Lotka-Volterra [Mur03] qui est le modèle le plus ancien mais également le plus reconnu pour calculer la dynamique des systèmes biologiques de type proie-prédateur. L'idée sous-jacente du modèle de Lotka-Volterra est que les populations de consommateurs (prédateurs) et de ressources (proies) peuvent être traitées comme des particules interagissant dans un liquide ou un gaz et, par conséquent, le taux de rencontre entre les consommateurs et les ressources serait proportionnelle au produit de leur masse. Les équations de Lotka-Volterra sont les suivantes :

$$\begin{cases} N(0) = N_0,\ P(0) = P_0, \\ \dfrac{\mathrm{d}N(t)}{\mathrm{d}t} = \alpha N(t) - \beta N(t)P(t) \\ \dfrac{\mathrm{d}P(t)}{\mathrm{d}t} = \delta P(t)N(t) - \gamma P(t) \end{cases}$$

avec :

- $N(t)$ la densité des proies, $P(t)$ la densité des prédateurs au temps $t \geq 0$, N_0 et P_0 étant les valeurs initiales des populations de proies et prédateurs.
- $\dfrac{\mathrm{d}N(t)}{\mathrm{d}t}$ décrit la croissance de la population des proies au cours du temps, α étant la croissance naturelle des proies sans prédation et β la mortalité des proies due aux prédateurs.
- $\dfrac{\mathrm{d}P(t)}{\mathrm{d}t}$ décrit la croissance de la population des prédateurs au cours du temps, δ étant la croissance des prédateurs en fonction des proies mangées et γ le taux de mortalité des prédateurs en absence de proies.

Nous utilisons ce modèle pour calculer les contraintes temporelles des différents états intermédiaires (invariant et garde) de l'évolution de la biomasse en fonction des types d'événements déclencheurs. Ce modèle est basé sur un système d'équations non-linéaires pour lequel nous avons proposé une résolution analytique, sous certaines hypothèses (possibles grâce à la discrétisation des niveaux de biomasse), nous permettant une résolution analytique moins coûteuse en temps et donc plus compatible avec un outil d'aide à la décision interactif [Zha14].

Génération. Pour construire le réseau d'automates, l'algorithme génère chaque automate représentant un sous-système de manière indépendante, la complexité de chacun dépendant du nombre d'états qualitatifs définis par l'utilisateur et du nombre d'interactions avec les sous-systèmes voisins. Pour chaque état intermédiaire des automates espèces, un appel au modèle de Lotka-Volterra est invoqué pour le calcul des contraintes temporelles. L'incertitude sur ces contraintes (valeurs des gardes et des invariants) dépend d'un intervalle de confiance donné par l'utilisateur sur ses paramètres en entrée. Dans certains cas de figure, les horloges nécessitent une remise à jour (lorsqu'une pression se sur-ajoute par exemple alors que la biomasse d'une espèce était déjà en cours d'évolution). Ces différentes mises en situation n'étaient pas toujours prévues par le formalisme des automates temporisés souvent appliqués à des cas d'évolution plus linéaires, et ont alors nécessité des ajustements un peu techniques mais néanmoins nécessaires pour que notre modèle ait du sens.

L'algorithme 2 décrit la construction automatique du réseau d'automates. Chaque automate est tout d'abord crée de manière indépendante. L'algorithme commence par construire les états stables et intermédiaires en fonction du voisinage (autres espèces, pressions, phénomènes climatiques) avant de rajouter les arcs et contraintes temporelles associées.

Simplification du modèle. La génération du modèle peut amener à un réseau d'automates de très grande taille. L'analyse des contraintes temporelles portées par les transitions menant aux états intermédiaires des espèces montre de nombreuses similitudes dans les valeurs des invariants et des gardes. L'idée consiste donc à réduire la taille des automates en détectant les états intermédiaires similaires et en les fusionnant. Ces états intermédiaires sont considérés comme similaires s'ils possèdent les même propriétés : même valeur qualitative portée par l'état, mêmes états but et origine pour les transitions entrantes et sortantes de ces états. L'intégrité du modèle reste garantie après la fusion des états, sa sémantique étant la même puisque seules les contraintes temporelles sont agrégées. Un algorithme de clustering est appliqué sur les états intermédiaires des espèces. La distance utilisée

Algorithm 2 Génération du modèle global de l'écosystème

Création d'un automate temporisé A_i pour chaque espèce
for all espèces **do**
1) Créer les états stables des N niveaux qualitatifs
2) Créer les N^j états intermédiaires pour les j automates voisins
for all états intermédaires **do**
a) créer les arcs entrants : événements reçus, remise à zéro des horloges
b) étiqueter les arcs entrants avec les événements possibles issus des voisins
c) *invariant*, *garde* ← Lotka-Volterra-Routine()
d) ajouter l'*invariant* à l'état
e) créer l'arc sortant et l'étiqueter avec la *garde* et l'événement émis
end for
end for
3) Créer les états *transversaux* entre les états intermédaires : mise à jour de l'horloge $x = update(x)$
4) Créer les arcs de retours possibles
Création d'un automates temporisé A_j pour chaque pression
for all pression de pêche et perturbation environnementale **do**
Créer les N niveaux qualitatifs donnés par les utilisateurs
Créer les contraintes temporelles définies par les utilisateurs
end for
Création du modèle global A
A = $\mathcal{A}_{i1} \,||\, \mathcal{A}_{in} \cdots ||\, \mathcal{A}_{j1} \,||\, \mathcal{A}_{jn}$

est la suivante : $d = \sqrt{\Delta I^2 + \Delta G^2}$ tel que ΔI (resp. ΔG) est la différence entre les invariants (resp. gardes sortantes) de ces états. L'état intermédiaire résultat de chaque classe a pour invariant et garde sortant, la moyenne des valeurs des invariants et gardes des éléments de sa classe. Il est possible de paramétrer le nombre d'états intermédiaires maximum entre deux états stables de biomasse. Des études sur des applications réelles ont montré que 12 états intermédiaires étaient un bon compromis pour maintenir la précision du modèle tout en garantissant des temps de réponse acceptables pour son exploration.

3.4.3 L'écosystème marin du lagon d'Ouvéa

Pour consolider ces travaux, j'ai proposé un séjour post-doctoral à Agrocampus Ouest, sur un thème pluridisciplinaire "écologie et modélisation qualitative" afin de pouvoir travailler sur un écosystème réel. Le séjour post-doctoral d'Yves-Marie Bozec a permis d'appliquer ce principe de modélisation sur l'écosystème récifo-lagonaire d'Ouvéa. Ouvea est un grand lagon ouvert sur l'océan appartenant aux

Iles Loyauté en Nouvelle-Calédonie. La pêche n'y est qu'une pêche de subsistance, les captures de pêches ont été estimées à partir des consommations de poissons et de fruits de mer par les habitants locaux à l'aide d'enquêtes [LFL04]. Yves-Marie Bozec avait déjà établi, durant sa thèse, une description quantitative de la structure des communautés, des relations trophiques et des pressions de pêches [BGK04].

Nous avons décidé de nous appuyer sur les données existantes pour définir un modèle d'automates temporisés de cet écosystème. Yves-Marie Bozec, qui est écologue marin et spécialiste des lagons, a sélectionné les cinq espèces les plus représentatives : les grands herbivores ($sp1$), les petits planctivores ($sp2$), les petits carnivores ($sp3$), les grands carnivores ($sp4$) et les piscivore ($sp5$) (cf. [BGK04] pour une description complète de ces groupes d'espèces). Alors que $sp1$ subit seulement la prédation de $sp5$, $sp2$ est consommée par les trois prédateurs $sp3$, $sp4$ et $sp5$ qui sont soumises à des pressions de pêches (cf. Figure 3.3). Cet écosystème est intéressant par les effets potentiellement non-intuitifs de la pêche sur $sp5$, le seul prédateur des deux espèces $sp3$ et $sp4$ qui eux aussi subissent une pression de pêche.

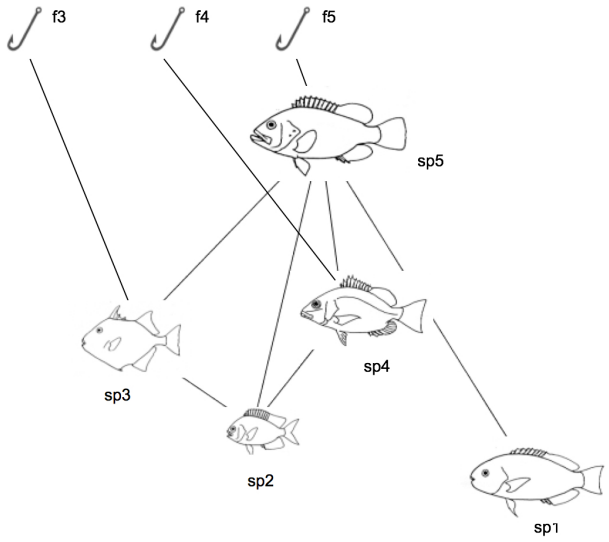


FIGURE 3.3 – Ecosystème marin d'Ouvéa (Nouvelle-Calédonie)

Pour chaque espèce, nous avons défini cinq niveaux qualitatifs : *normal*, *low*, *high* et *extinct*. Le niveau normal correspond à la valeur initiale de la biomasse (appelée valeur à l'équilibre dans les modèles écologiques), *low* et *high* sont respectivement la moitié et le double de la valeur *normal* et *extinct* correspond à une valeur quasi nulle (soit 5% de la valeur normale). Les pressions de pêche sont

données à partir des données collectées sur le terrain. Le niveau *normal* correspond à la pêche actuelle, alors que les niveaux *high* et *low* ont été définies spécifiquement pour chaque espèce. Concernant les informations temporelles, l'unité de temps définie est le mois et l'horizon temporel est fixé à 20 ans. La variabilité apposée sur les données en entrée génère une incertitude sur les contraintes du temps de passage entre les différents niveaux de biomasse des espèces. La publication [LCB+11] donne plus de détails sur ce modèle. La génération automatique du modèle global ne prend que quelques secondes (sur un Apple Mac Book classique). L'automate de l'espèce *sp*5 est le plus complexe étant données ses nombreuses interactions. Le modèle final contient 1689 états et 11552 transitions.

Nous aborderons dans les sections suivantes l'exploration de ce modèle selon les différents types de stratégies ainsi que la validation de la modélisation en comparant les résultats proposés par les requêtes avec des données historiques.

Bilan sur la modélisation d'écosystèmes

Cette étude, bien que présentée ici de manière linéaire, nous montre toute la difficulté de proposer, dans un contexte pluridisciplinaire, des méthodes de modélisation. Ces méthodes doivent tenir compte à la fois des exigences informatiques : unité du formalisme employé, généralisation, réduction de la complexité, efficacité, et de leur application sur des écosystèmes réels. Les chercheurs en écologie sont pour leur part très spécialisés, sur un type d'écosystème donné (voire une ou deux espèces) dans des zones géographiques précises. Leur compétence tient à la connaissance qu'ils possèdent de toutes les spécificités de leur domaine d'étude. Il nous a donc fallu trouver un juste milieu, et de nombreux compromis, afin de proposer une vision de modèle *holistique* permettant de formaliser à la fois les espèces les plus significatives et les différentes pressions de pêche. Les différentes questions ont porté sur le choix des niveaux qualitatifs, l'unité de temps la plus adaptée, la prise en compte des pressions multiples (et par conséquent la mise à jour des horloges), l'évolution de la valeur de la biomasse (et le recours au modèle de Lotka-Volterra), la définition des pressions de pêche et de leur niveau. Sans un investissement fort de la part de nos experts (Yves-Marie Bozec à temps plein pendant une année et Guy Fontenelle en soutien hebdomadaire sur une période de cinq ans), ce travail de modélisation n'aurait pu aboutir. Depuis 2010, début de cette étude, les choses ont bien évolué et les modèles qualitatifs, rares à l'époque dans le domaine halieutique (en dehors des travaux de Jeffrey Dambacher au CSIRO en Australie qui fut d'un fort soutien), commencent à se développer dans le soucis d'intégrer la dimension humaine sous ses aspects sociologiques, économiques et culturels, dans la représentation des écosystèmes.

3.5 Requêtes prédictives *what-if ?* par model-checking

Nous introduisons dans cette section une première approche pour explorer le modèle de l'écosystème à l'aide de requêtes. Nous nous plaçons ici dans le cadre des scénarios prédictifs ou *what-if* (que se passe-t-il si...) qui proposent *une description d'une situation future [...] basée sur des hypothèses spécifiques sur le futur, et (lorsque cela est approprié) incluant également la présentation du développement du présent vers ce futur* [PEF00]. Dans un écosystème, les *hypothèses spécifiques sur le futur* correspondent aux unités environnant le système biologique : les actions anthropiques et les perturbations. L'idée consiste donc à poser un jeu d'hypothèses, à construire le modèle global et à appliquer des requêtes afin d'envisager le futur sous ces conditions de pressions. De par la nature qualitative du modèle, et l'échéance temporelle qui est du moyen ou long terme, nous considérons que nous nous situons dans le spectre des *méthodes exploratoires* dont l'objectif est surtout la projection de tendances.

Les techniques de model-checking ont l'intérêt de s'appuyer sur un formalisme logique qui permet de proposer un panel de requêtes assez étendu. Bien que le cadre logique offre de nombreuses possibilités, nous avons préféré guider les gestionnaires dans leurs interrogations, afin de leur éviter d'avoir à manipuler directement la logique TCTL, en leur proposant des patrons de requêtes facilement paramétrables.

3.5.1 Patrons de requêtes

Les patrons de scénarios que nous avons identifiés reposent, pour la plupart, sur la notion de *situation*. Une situation S représente un état global de l'écosystème (ES) et est représenté par un sous-ensemble des états possibles de ses sous-modèles.

SITUATION
Etant donné un écosystème décrit par réseau d'automates temporisés composé de n automates, dont m automates A_i pour l'écosystème naturel ES ($m < n$), une situation S décrivant ES est : $S = p_{A_1} \wedge p_{A_2} \wedge \cdots p_{A_n}$ avec p_{A_i} une disjonction des propriétés possibles de A_i ($i \in [1, m]$).

Le cas particulier de S_{init} affecte à toutes les espèces de l'écosystème la valeur initiale de leur biomasse. Un exemple de situation peut être $S = (sp1_low \vee sp1_extinct) \wedge sp2_normal \wedge sp3_high$. On parle de Sc pour une situation complète au sein de laquelle toutes les espèces de l'écosystème sont évaluées.

Avec l'aide des experts, nous avons identifié six patrons de requêtes qui permettent de répondre aux questions les plus couramment posées sur un écosystème.

Notre contrainte a été leur expressibilité dans la logique temporelle TCTL. La figure 3.4 illustre ces différents patrons.

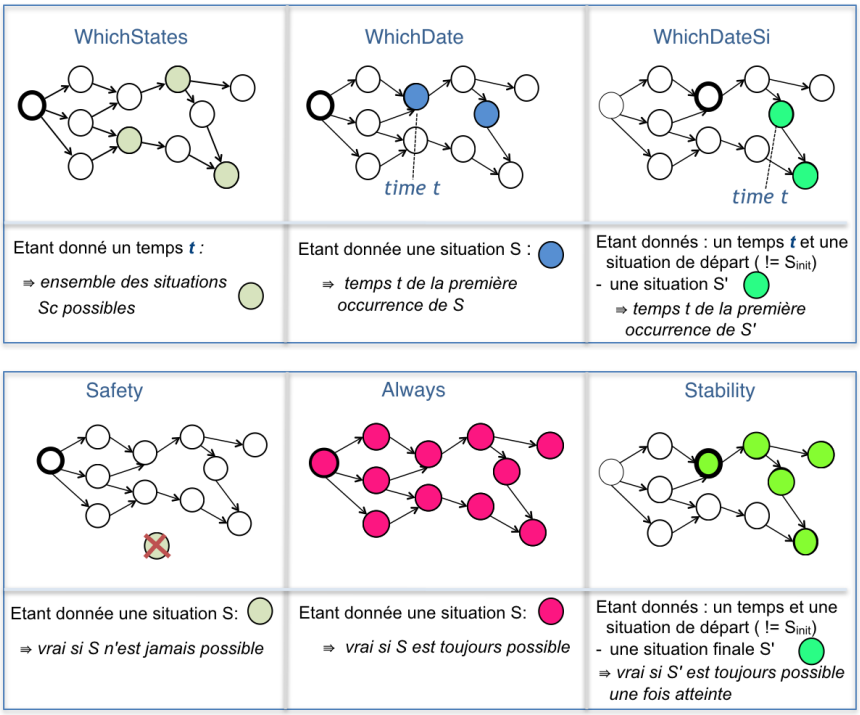


FIGURE 3.4 – Six patrons de requêtes pour explorer un écosystème

Le patron *WhichStates* est utilisé pour rechercher toutes les situations finales possible Sc, à une date donnée t, suite à l'application d'une politique d'exploitation depuis la situation de départ S_{init} (avec ou sans prise en compte d'événement lié à l'environnement). Ce type de scénario est surtout utilisé pour comparer les impacts des différentes politiques d'usage à une échéance plus ou moins longue que définit le gestionnaire. Le patron *WhichDate* permet de connaître le temps de réaction de l'écosystème pour atteindre une situation S suite à l'application de pressions, en partant de la situation S_{init}. La variante *WhichDateSi* calcule ce temps de réponse en partant d'une situation donnée, elle répond au besoin de connaître le temps de réaction d'un écosystème si l'on modifie la politique appliquée. Ce patron permet, en particulier, de tester une propriété intéressante en écologie : la résilience d'un l'écosystème (sa capacité à retrouver une situation normale si l'on supprime toute forme de pression). Le patron *Never* répond au problème classique des requêtes dites de sûreté en s'assurant qu'une situation à risque (une espèce qui atteint un niveau relatif à son extinction par exemple) ne peut jamais être atteinte avec les

hypothèses posées. Son homologue est *Always* qui permet de garantir des niveaux satisfaisants pour des espèces ciblées. Enfin le patron *Stability* peut s'avérer utile afin de vérifier qu'une situation reste stable (et ne peut empirer en particulier) avec la politique d'exploitation appliquée. La syntaxe TCTL des patrons de requêtes est détaillée dans la table 3.1 [LCB+11]

Les requêtes reposent sur des propriétés TCTL suivantes :
— Atteignabilité (de type $\exists\Diamond S$) pour *WhichStates*, *WhichDate*, *WhichDateSi*
— Invariance (de type $\forall\Box S$) pour *Never*, *Always*
— Vivacité (de type $\forall\Box(S => \forall\Diamond S')$ pour *Stability*

Patrons de requête	Description / Entrées	Expression.
WhichStates	Recherche les situations possibles à t - une date t	for all S_i : $\exists\Diamond(S_i \wedge chrono.t = t\ )$
WhichDate	Recherche la date t_i de première occurrence de S - situation finale S	for all $t_i \in [0, t_{max}]$ $\exists\Diamond(S \wedge chrono.t = t_i)$
WhichDateSi	Recherche la date t_i de la première occurrence de S' partant de S - situation initiale $S(\neq S_{init})$ - date de départ t - situation finale S'	if $((\exists\Diamond(S \wedge chrono.t = t) == true)$ for all $t_i \in [0, t_{max}]$ $(S \wedge chrono.t = t) \Rightarrow \exists\Diamond(S' \wedge chrono.t = t_i)$
Safety	Vérifie que la situation S ne se produit jamais - situation S	$not(E\Diamond S)$ or $\forall\Box(notS)$
Always	Vérifie qu'une situation S est toujours vérifiée - situation S	$\forall\Box(S)$
Stability	Recherche une situation stable S - situation S	$A\Box(S \Rightarrow A\Box S)$

TABLE 3.1 – Les six patrons de requêtes et leur expression TCTL

Ces patrons diffèrent par leur type de retour. Le model-checking retourne des résultats de type *true/false* qui ne sont pas nécessairement suffisants face aux attentes des gestionnaires. Bien qu'ils s'avèrent compréhensibles pour les trois derniers patrons *Never*, *Always* ou *Stability*, se limiter à des requêtes proposant uniquement des réponses binaires était inenvisageable pour un outil d'aide à la décision. Ainsi, les trois premiers patrons recherchent soit un ensemble de situations accessibles, soit une date. Ces attendus sont plus de l'ordre d'un résultat de *simulation* d'un modèle classique que d'une vérification par model-checking. Pour y parvenir, nous avons dû englober ces requêtes au sein d'un processus de recherche qui effectue un ensemble de vérifications (le plus petit possible) afin de fournir des informations agrégées (pour les situations finales) ou la date de la première occurrence d'une situation. Par ailleurs, l'incertitude temporelle du modèle sur l'évolution des espèces peut mener à plusieurs situations finales. Pour le patron *WhichStates* il est également possible de découvrir les trajectoires menant aux états finaux à l'aide de vérifications plus fines. Si la biomasse d'une espèce est en cours d'évolution, ses deux états qualitatifs possibles à la date d'interrogation

figurent alors dans la solution proposée. Ce choix est une volonté afin de rendre le modèle plus facilement interprétable pour les gestionnaires.

3.5.2 Logiciel EcoMata

La construction automatique du modèle comme les patrons de requêtes ont été implémentés au sein du logiciel EcoMata. Ce travail a été réalisé par Yulong Zhao durant la première partie de sa thèse. EcoMata [1] est un outil d'aide à la décision destiné aux gestionnaires d'écosystèmes marins afin de leur permettre de tester l'impact des pressions de pêches et des événements climatiques sur les espèces [YZC11].
Il est composé de trois parties (cf. figure 3.5) :

- L'*éditeur de réseau* qui offre des outils graphiques pour dessiner le graphe des espèces, les pressions de pêches sous la forme de chronogrammes ainsi que les événements climatiques.
- Le *générateur d'automates* qui invoque l'algorithme de construction du modèle global présenté plus haut.
- Le *lanceur de requêtes* qui propose une interface pour paramétrer les patrons de requêtes.

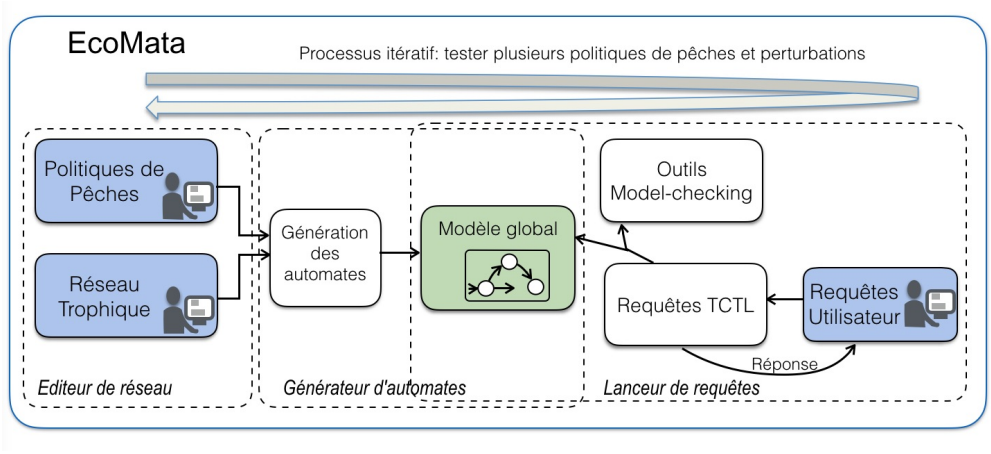


FIGURE 3.5 – Logiciel EcoMata

EcoMata s'appuie sur l'outil de model-checking UPPAAL [LPY97] pour la vérification des requêtes. L'objectif est le développement d'un outil simple d'utilisation qui propose des résultats rapidement afin de pouvoir envisager son usage

1. EcoMata peut être téléchargé à partir de la page web suivante : https ://team.inria.fr/dream/fr/ecomata

au sein de réunions de concertation entre gestionnaires où différentes hypothèses d'exploitation et d'événements climatiques pourraient être évaluées dans un processus itératif. La figure 3.6 présente un exemple de résultat fourni par EcoMata pour le patron *WhichStates*, les deux situations possibles à la date demandée sont présentées dans la fenêtre du haut, tandis que les trajectoires pour chaque espèces sont présentées dans celle du bas.

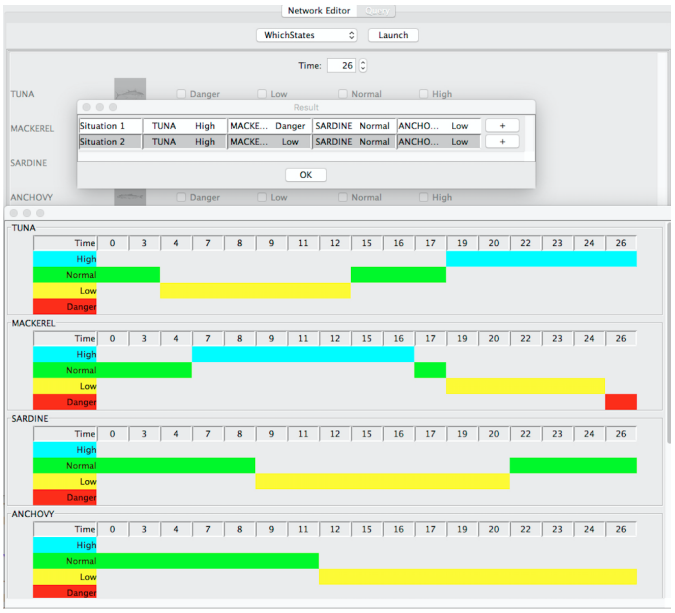


FIGURE 3.6 – Résultat du patron *WhichStates* à l'aide d'EcoMata

Des benchmarks ont été effectués afin de pouvoir classifier le modèle généré selon sa complexité et ainsi informer l'utilisateur des délais escomptés. EcoMata peut être utilisé avec des délais raisonnables avec pour limite un écosystème définissant 6 pressions de pêches et 7 espèces définies par 4 niveaux qualitatifs chacune, chaque espèce ayant 3 interactions au maximum.

3.5.3 Expérimentations

Des expérimentations ont été menées avec pour chacun d'elles différents objectifs :

— L'écosystème marin d'Ouvéa pour tester les patrons de requêtes sur trois scénarios de pêches réalistes : une exploitation constante et forte, une exploitation périodique et forte et un scénario alternatif sur la force de la pêche. Cette expérimentation a été menée conjointement avec Yves-Marie Bozec et Guy Fontenelle [LCB+11].

— L'écosystème de la mer du Nord en se plaçant dans un contexte rétroactif pour construire le modèle et comparer les prédictions fournies par EcoMata avec des données de captures disponibles sur cette zone, dans un objectif de validation de l'approche [PFGL11]. Cette étude a été menée par Charlotte Paillette durant un stage de master 2 en co-encadrement avec Didier Gascuel du laboratoire Halieutique d'Agrocampus Ouest [Pai11].
— L'écosystème de la coquille Saint-Jacques en rade de Brest avec pour objectif la gestion de son stock. Ce travail a été réalisé au sein du projet "RAD2BREST Modélisation et scénarisation des activités humaines en rade de Brest" coordonné par François Gourmelon du laboratoire Géomer de l'Université de Bretagne Occidentale [PCL+11].
— Un écosystème de type adventices/carabes dans un paysage agricole avec pour objectif la diminution des adventices présentes sur les culture tout en limitant l'usage des pesticides (par la présence de carabes). Outre le domaine d'application qui sort du champ des écosystèmes marins, il s'agit d'étendre la représentation à des processus spatiaux. Cette étude a été réalisée par Pierre Gaignon durant son stage de master 2 en co-encadrement avec Sylvain Poggi et Nicolas Parisey de l'équipe IGEPP de l'INRA du Rheu [Gai15]

3.5.4 Bilan sur l'exploration de modèles par des scénarios prédictifs

La modélisation qualitative se situe à un niveau d'abstraction plus élevée que les approches par modèles numériques et est bien adaptée au cas où des utilisateurs veulent comparer plusieurs scénarios à moyen ou long terme (20 ans) et prendre des décisions à partir des informations issues de l'exploration des modèles. L'exploration des modèles sur les expérimentations a confirmé l'usage de cette approche en tant qu'indicateur de tendance. Ainsi, sur le site d'Ouvéa, nous avons pu remarquer que les prévisions montraient une mise en danger d'une espèce après 14 années d'exploitation intensive alors qu'une pêche périodique laissait du temps à l'espèce pour se renouveler entre deux périodes de pêche. La modélisation a pu s'appliquer sans difficulté à un écosystème de coquilles Saint-Jacques qui a été exploré selon différentes stratégies de pêches avec une intensité associée au nombre de bateaux envisagés. Les patrons de requêtes se sont révélés utiles par la variété des interrogations sur ce domaine. L'écosystème de la mer du Nord nous a permis de comparer l'évolution des stocks prédit par la modélisation par rapport à des données collectées sur vingt ans. Le modèle bien que complexe (sept espèces et sept pressions de pêche) a pu être analysé par tous les patrons de requêtes disponibles (jusqu'à cinq minutes d'exécution au maximum pour les requêtes les plus

complexes). Bien que les données elles-mêmes soient sujettes à des imprécisions (les valeurs de biomasse étant issues des données de captures et non d'une réelle observation), cette étude a permis de confirmer que les valeurs qualitatives des biomasses des espèces prédites par EcoMata étaient cohérentes avec les données. L'étude sur la mer du Nord a cependant mis en avant l'impossibilité d'intégrer une variation importante et subite des biomasses par un afflux de population provenant d'une autre zone. Effectivement, les dynamiques spatiales n'ont pas été prises en considération dans la modélisation EcoMata, d'une part parce que traiter à la fois les dimensions temporelle et spatiale est un problème bien connu pour sa complexité, d'autre part parce que les écologues eux-mêmes travaillent sur des zones bien circonscrites (lagon, rade, etc.), et le besoin n'avait pas encore été exprimé. Nous avons cependant abordé le problème avec l'expérimentation de l'écosystème de type adventices/carabes. Ici, les espèces ont été limitées à une seule interaction proie(adventices)/prédateur(carabes) sur un territoire compartimenté au sein duquel les migrations ont également été modélisées. La complexité des entités (automates de paysage, automate de population, automate de migration et de diffusion des graines) génère un modèle exponentiel avec le nombre de parcelles envisagées. La plupart des requêtes n'ont pu aboutir par manque de mémoire (limité à 2Go par UPPAAL, l'usage d'un serveur ne pouvant être d'aucune aide) et aucune simplification n'est apparue possible sans perdre en cohérence. Bien qu'intéressante du point de vue de la modélisation, en particulier la représentation des phénomènes spatiaux temporels par automates, cette étude a confirmé la difficulté d'envisager une modélisation à la fois temporelle et spatiale pour une interrogation par model-checking.

Les scénarios prédictifs sont intéressants comme outil d'aide à la décision puisqu'ils engagent directement les gestionnaires qui définissent eux-mêmes les scénarios d'exploitation. Ces gestionnaires sont également actifs dans le jeu des requêtes qui les invitent à se projeter dans le futur. Des allers et retours entre les hypothèses posées (les actions humaines, les événements environnementaux) et les prévisions devraient leur permettre de converger vers une politique d'exploitation satisfaisante, à la fois pour les usagers de l'écosystème, que pour l'écosystème lui-même en terme de durabilité. Cependant, il peut s'avérer que la situation nécessite une aide plus importante du système, avec la proposition de politiques de pêche.

3.6 Requêtes proactives *what-for ?* par synthèse de contrôleur

Si le champ d'exploitation est plus large, les scénarios prédictifs *what-if ?* ne sont pas toujours suffisants pour permettre aux gestionnaires d'identifier les politiques d'exploitation les plus appropriées. Nous nous sommes donc intéressés au problème de recherche de stratégies optimales permettant d'atteindre un objectif. Ces requêtes sont appelées *what-for ?* et traduisent la question *Que faut-il faire pour atteindre l'objectif ?* Pour les spécialistes des scénarios ayant défini une topologie, ces questions appartiennent au cas des scénarios *normatifs* dont la sous-catégorie, les *scénarios de préservation*, concernent particulièrement les écosystèmes que l'on souhaite protéger [BHD+06].

L'approche proposée par ce travail repose sur la synthèse de contrôleur des automates temporisés. Nous proposons de répondre à la question : *Quelle est la meilleure stratégie pour guider le système vers un but précis à une date donnée ?* Nous verrons qu'utiliser la synthèse de contrôleur nécessite de redéfinir les automates temporisés afin de distinguer les actions contrôlables des actions non contrôlables et d'ajouter la notion de coût.

Dans un premier travail, nous avons employé la synthèse de contrôleur pour la recherche de stratégies sans avoir recours à la notion de coûts. Sur un modèle aussi conséquent qu'un écosystème, la synthèse de contrôleur menait à un contrôleur trop important en nombre de stratégies qu'il était difficile d'exploiter par la suite comme outil d'aide à la décision. Nous avons montré qu'une méthode de type "générer et tester" utilisant a posteriori une fonction d'évaluation était dans ce cadre plus efficace [ZCL12].

Nous présentons donc ici le formalisme des PTGA (Priced Timed Game Automata) une extension des automates temporisés et les méthodes employées pour la recherche de stratégies.

3.6.1 Automate temporisé de jeu et à coûts

Les automates de jeu (TGA pour *Timed Game Automata*) [AMP95] étendent les automates temporisés avec des actions qui sont partitionnées en deux : les actions *contrôlables* et les actions *non contrôlables*. Un TGA peut être utilisé pour représenter un modèle à deux joueurs entre un agent que l'on peut appeler un contrôleur et l'environnement. Les automates à coûts (PTA pour *Priced Timed Automata*) [ALTP01] étendent également le formalisme des automates pour rajouter des coûts sur le comportement du système. Le coût sur le sommet de l'automate représente alors le coût par unité de temps resté dans le sommet alors que le coût sur l'arc représente le coût de la transition lorsque celle-ci est déclenchée. Partant

de ces principes, l'automate temporisé de jeu à coûts (PTGA pour *Priced Timed Game Automata*) est une extension à la fois des PTA et TGA [BCFL04].

La figure 3.7 présente un PGA de quatre sommets : $S1$, $S2$, $S3$ et *Goal* avec $S1$ comme sommet initial. Les transitions en traits pleins représentent les actions contrôlables ($a1$, $a2$, $a3$) et les transitions en pointillés représentent les actions non contrôlables ($u1$, $u2$). Le PTGA définit une horloge x pour la définition des invariants et des gardes (en vert). La notion de *coût* est exprimée par *cost*′ sur les sommets indiquant le coût par unité de temps et *cost* sur les arcs représentant le coût de la transition. La notion de *jeu* est représentée par le but, le sommet *Goal* ainsi que par les actions non contrôlables $u1$ et $u2$.

Automate temporisé de jeu et à coûts (PTGA *Priced Timed Game Automata*) [BCFL04]
Un PTGA, automate temporisé de jeu et à coûts, A est défini par un n-uplet $< \mathcal{S}, \mathcal{X}, \mathcal{A}ct, \mathcal{E}, \mathcal{I}, \mathcal{P} >$ avec :

- $\mathcal{S}$ est un ensemble fini de sommets, $s_o \in S$ étant le sommet initial, $Goal \subseteq S$ définit les états buts ;
- $\mathcal{X}$ est un ensemble fini d'horloges ;
- $\mathcal{A}ct = Act_c \cup Act_u$ est un ensemble fini d'actions divisé en deux ensembles : Act_c, les actions *contrôlables* (jouées par le contrôleur), et Act_u, les actions *non contrôlables* (jouées par l'environnement) ;
- $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{A}ct \times \Phi(\mathcal{X}) \times 2^{\mathcal{X}} \times \mathcal{S}$ est l'ensemble fini des arcs. Chaque arc e est un n-uplet $(s, a, \varphi, \delta, s')$ tel que e relie le sommet $s \in \mathcal{S}$ au sommet $s' \in \mathcal{S}$ par l'action $a \in \mathcal{A}ct$. La condition d'activation (appelée la *garde*) est représentée par $\varphi \in \Phi(\mathcal{X})$. $\delta \subseteq \mathcal{X}$ donne l'ensemble des horloges réinitialisées lorsque la transition est tirée.
- $\mathcal{I} : \mathcal{S} \to \Phi(\mathcal{X})$ associe chaque sommet s à une contrainte d'horloge appelée un *invariant*.
- $\mathcal{P} : \mathcal{S} \cup \mathcal{E} \to \mathbb{N}$ assigne des coûts aux sommets et des taux de coûts aux transitions respectivement.

La sémantique d'un PTGA est défini comme *un système de transitions à coûts* pour lequel les états $q \in \mathcal{Q}$ sont des paires $q = (s, v)$ tel $s \in \mathcal{S}$, est un sommet, et v une valuation d'horloge où v satisfait l'invariant $\mathcal{I}(s)$ du sommet s. On note $q \xrightarrow{a}_c q'$ une transition discrète entre deux états q et q' étiquetée par l'action a alors que c est le coût de la transition.
Une transition $(s, v) \xrightarrow{\Delta}_c (s, v')$ est une transition temporelle si $c = \Delta \times \mathcal{P}(s)$, $v' = v + \Delta$ et l'invariant I(s) est satisfait par l'état source, l'état cible et tous les états intermédiaires tels que $v + \Delta' \models \mathcal{I}(s)$ avec $\Delta' \leq \Delta$.

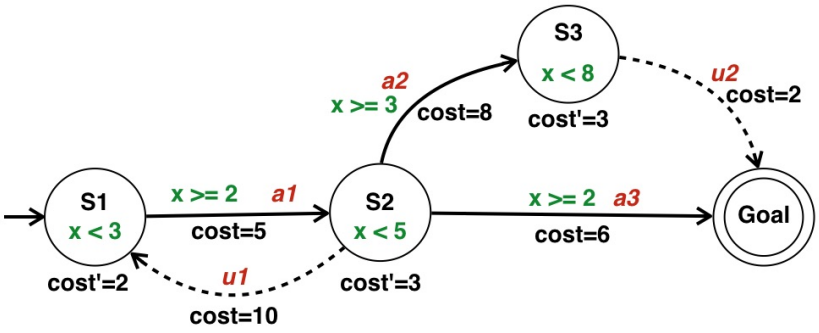


FIGURE 3.7 – Exemple de PTGA

Trajectoire Une trajectoire d'un PTGA est une séquence $q_0 \xrightarrow{\Delta_1}_{c_1} q_1 \xrightarrow{a1}_{c_2} q_2 \xrightarrow{\Delta_2}_{c_3} q_3 \xrightarrow{a2}_{c_4} q_4 \cdots$ alternant transitions temporelles et transitions discrètes. Une trajectoire d'un PTGA est un chemin dans le système de transition à coûts.

Soit ϱ est une trajectoire finie, le coût de ϱ, noté $cost(\varrho)$ est la somme de tous les coûts le long de cette trajectoire telle que $cost(\varrho) = \sum_i c_i$. Par exemple, si l'on considère le PTGA de la figure 3.7, une trajectoire ϱ, non-optimale, entre l'état initial et *Goal* peut être la suivante. Pour les transitions temporelles, le temps passé au sein de chaque sommet est indiqué entre crochet. Le coût de ϱ est de 21.

$$\varrho : (s_1, 0) \xrightarrow{\Delta_{[2]}} (s_1, 2) \xrightarrow{a1} (s_2, 2) \xrightarrow{\Delta_{[2]}} (s_2, 4) \xrightarrow{a3} (Goal, 4)$$

3.6.2 Synthèse de contrôleur

La théorie générale de la supervision des SED par la synthèse de contrôleur a été introduite par P. J. Ramadge et W. M. Wonham(R&W) [RW87]. L'idée de l'approche R&W consiste à synthétiser un contrôleur pour un modèle donné, tel que ce contrôleur couplé au modèle de fonctionnement reste toujours dans l'ensemble des comportements autorisés, tout en restant le plus permissif possible. De nombreuses extensions ont été proposées, en particulier pour les systèmes temporisés [AMPS98].

Dans un PTGA, on appelle Σ l'ensemble des actions possibles que peut proposer le contrôleur tel que $\Sigma = Act_c \cup \lambda$ avec Act_c l'ensemble des actions contrôlables et λ l'action d'attente (laisser passer le temps). Seules les actions contrôlables peuvent être contrôlées par le contrôleur. L'interaction entre le système et le contrôleur

peut être vue comme un jeu à deux joueurs. Dans un tel jeu, une *stratégie* est un ensemble de *règles de décision* qui indique au contrôleur quelle action choisir. Une stratégie est gagnante si, en suivant ces règles de décision, il gagne toujours quelles que soient les actions de l'environnement (par le biais des actions non contrôlables).

Règle de décision
Une règle de décision définit une action qui peut être opérée à partir d'un sommet de l'automate temporisé pour une contrainte temporelle donnée. Une règle de décision est un tuple (s, φ, σ) tel que $s \in \mathcal{S}$ est un sommet de l'automate, $\varphi \in \Phi(\mathcal{X})$ une contrainte d'horloge et $\sigma \in \Sigma$ une action.

Une stratégie décrit le comportement du contrôleur, soit la séquence d'actions à opérer.

Stratégie
Dans un PTGA $\mathcal{A}$, une stratégie f est une relation partielle entre l'ensemble des états de l'automate $\mathcal{Q}$ et l'ensemble des actions Σ, $f \subseteq \mathcal{Q} \times \Sigma$. On peut dire qu'une stratégie f est un ensemble de règles de décisions.

Contrôleur
Un contrôleur C_f sur un PTGA appelé $\mathcal{A}$ est un système couplé avec $\mathcal{A}$ tel qu'il le contrôle selon une stratégie f. On note $C_f \mid\mid \mathcal{A}$, le PTGA $\mathcal{A}$ contrôlé par C_f.

Le problème de la synthèse de contrôleur sur un automate de jeu $\mathcal{A}$ peut être défini de la façon suivante : étant donné une propriété Φ, existe-il un contôleur C_f tel que $C_f \mid\mid \mathcal{A} \models \Phi$? La synthèse de contrôleur peut traiter deux types de problèmes décrits par Φ : l'*atteignabilité* et la *sûreté* :

— *Atteignabilité :* Etant donné un automate de jeu $\mathcal{A}$ et une propriété φ à atteindre, la synthèse de contrôleur consiste à trouver une stratégie f tel que $(C_f \mid\mid \mathcal{A}) \models \forall\Diamond\varphi$.

— *Sûreté :* Etant donné un automate de jeu $\mathcal{A}$ et une propriété φ à éviter, la synthèse de contrôleur consiste à trouver une stratégie f tel que $(C_f \mid\mid \mathcal{A}) \models \forall\Box\neg\varphi$.

Si la stratégie f n'est pas vide, le contrôleur C_f existe. Plusieurs stratégies peuvent potentiellement satisfaire la propriété φ à vérifier. On appelle $C^* = \{C_f \mid (C_f \mid\mid \mathcal{A}) \models \Phi\}$ l'ensemble possible des contrôleurs. Nous pouvons commenter deux types de contrôleur particulier.

— *Le contrôleur complet :* Le contrôleur complet $C_{fmax} \in C^*$ correspond à la stratégie complète qui contient toutes les règles de décision. Le contrôleur C_{fmax} est relié à la stratégie complète f_{max} telle que $\forall C_f \in C^*, f \subseteq f_{max}$.

Dans une stratégie complète, il est possible que, pour un même état de l'automate, plusieurs actions soit possibles.

— *Le contrôleur minimal :* Le contrôleur minimal C_{fmin} correspond à la *stratégie minimale* telle que : $\neg\exists C_f \in C^* \mid f \subset f_{min}$. Une stratégie minimale est un ensemble minimal de règles de décision. Dans ce cas, une seule action est possible pour un état de l'automate. La stratégie minimale n'est pas nécessairement unique.

Le coût d'une stratégie f sur un automate temporisé à coûts $\mathcal{A}$ depuis (s, v) est défini par :

$$cost((s, v), f) = sup\{cost(\rho \mid \rho \in outcome((s, v), f)\}$$

avec $outcome((s, v), f)$ un sous-ensemble de toutes les exécutions partant de (s, v) satisfaisant f [BCFL04].
Lorsque l'on s'intéresse au contrôleur complet le coût optimal de la stratégie depuis (s,v) est calculé de la manière suivante :

$$optCost(C_{fmax}, (s, v)) = inf\{cost((s, v), f) \mid f \in C_{fmax}\}$$

Outils
Aujourd'hui, une seule ligne de logiciels permet de manipuler les automates temporisés étendus, il s'agit Uppaal [LPY97]. Uppaal que nous avons déja utilisé pour le model-checking a été complété par deux autres logiciels : Uppaal Tiga [BCD+07] pour la synthèse de contrôleur sur des automates de jeu (TGA) et Uppaal Cora [BLR04] pour le model-checking sur des automates temporisés à coûts. Lors de la synthèse de contrôleur avec Uppaal Tiga, celui-ci peut retourner le contrôleur complet ou l'un des contrôleurs minimaux, tiré au hasard parmi tous les contrôleurs minimaux possibles. L'outil Uppaal Cora permet de calculer le coût de la trajectoire optimale.

3.6.3 Planification de stratégies

Les TGA représentent un formalisme intéressant pour la modélisation des écosystèmes puisqu'ils incorporent les notions inhérentes à ces systèmes comme la contrôlabilité ou non des actions et permettent la formulation des coûts utiles pour la valuation des actions anthropiques. La complexité de ces systèmes ne permet pas aujourd'hui d'appliquer la synthèse de contrôleur directement sur un système représenté par ce formalisme. Il est possible de calculer le coût d'une trajectoire optimale dans un problème d'atteignabilité sur un PTA ou de calculer un contrôleur pour un TGA. Ce contrôleur cependant, par nos expériences passées est

trop conséquent pour être exploité comme outil d'aide à la décision.

Notre problème consiste à trouver la meilleure stratégie, décrite comme un ensemble de règles de décision, permettant d'atteindre un objectif. Ce problème rejoint le domaine de la planification en IA, en particulier la planification temporelle pour laquelle quelques approches reposent également sur le model-checking [GT99, CPRT03, KMH00, CFF$^+$10, OFCF11, BHO14]. Le système peut être décrit sous la forme d'un PTGA puis dérivé en PTA ou TGA. Un algorithme *BestStratSearch* reposant sous les outils de model-checking d'Uppaal a été proposé afin de calculer un contrôleur complet pour la stratégie optimale.
Les principes de l'algorithme sont les suivants (décrits dans [LKZ16]) :

— Dériver le PTGA en PTA afin de calculer le coût *optCost* de la trajectoire optimale menant à un but *Goal* sur un problème d'atteignabilité : $\exists \diamond Goal$
— Dériver le PTGA en TGA en intégrant le coût au sein des variables associées aux sommets et arcs dans le formalisme d'Uppaal. Appliquer la synthèse de contrôleur sur la formule $\exists \diamond Goal \wedge \gamma$ (γ posant une contrainte $optCost$) afin d'obtenir le contrôleur complet de la trajectoire optimale.

L'intérêt de la méthode réside dans la réduction de la taille du contrôleur qui bien que complet se focalise sur la trajectoire optimale du problème d'atteignabilité. Cette résolution est plus efficace que de recalculer ce coût sur le contrôleur représentant toutes les trajectoires possibles puisqu'il s'appuie en amont sur des méthodes symboliques de model-checking. La dérivation du PTGA en PTA nécessite cependant une hypothèse forte : le non-déterminisme ne peut s'appliquer que sur les actions contrôlables du modèle de départ. Cette contrainte permet de garantir le coût de la stratégie optimale calculée sur le PTA [BCFL04].

Nous avons appliqué cette approche à un problème classique de la planification, un système de logistique appelé *Transport Domain* dont des jeux de données sont disponibles sur le site de l'IPC (International Planning Competition). Le système représente un ensemble de camions de diverses capacités qui doivent délivrer des colis dans différentes destinations. Le problème de planification est le suivant : *Etant donné un état initial où les camions et les colis sont à leur point de départ, quelle est la meilleure stratégie permettant aux colis d'atteindre leur lieu de livraison avec une consommation de carburant optimale ?*. Nous avons traduit le problème sous un réseau de PTGA (camions, colis, dépanneur), ajouté des contraintes temporelles par rapport au problème initial et ajouté une action incontrôlable qui est la *panne* du camion nécessitant l'intervention d'un dépanneur. L'algorithme *BestStratSearch* a été appliqué sur une série d'expérimentations du benchmark définissant un nombre croissant de points de livraison, de camions et de colis. Notre approche a été comparée avec un MCMAS-SLK un model-checker pour

la vérification de systèmes multi-agents [ČLMM14]. Les résultats montrent que *BestStratSearch* propose de meilleures performances et une plus grande capacité à traiter des systèmes de grande taille que MCMAS-SLK qui atteint l'explosion combinatoire rapidement [LKZ16].

PaturMata
Dans le domaine de l'agro-écologie, cette approche a été appliquée à la recherche de stratégies optimales de la gestion du pâturage. La gestion de pâturage nécessite la compréhension des interactions entre prairies et troupeaux dans le but d'optimiser l'usage de la prairie tout en limitant l'impact environnemental. Cette étude, réalisée dans la thèse de Yulong Zhao [Zha14], a donné lieu à l'outil d'aide à la décision PaturMata qui propose les stratégies optimales des activités de paturage d'une exploitation agricole : la mise au pâturage, le fauchage, la fertilisation. Le modèle de PaturMata diffère de celui d'EcoMata sur deux points : i) il est organisé comme un modèle hiérarchique, ii) c'est un modèle hybride combinant un modèle numérique (croissance de l'herbe) et des modèles qualitatifs représentés par des automates de type PTGA. Le modèle hiérarchique est constitué d'une couche basse avec le modèle numérique de croissance de l'herbe, une couche d'exécution représentant les actions unitaires (coupe de l'herbe, fertilisation) qui peuvent être appliquées sur la prairie et enfin la couche supérieure définissant les stratégies de management. Des expérimentations ont été menées sur un troupeau de cinquante animaux et une exploitation constituée de sept prairies pour laquelle on cherche la stratégie de fertilisation optimale. La fonction de coût prend en compte la consommation de fourrage et le prix de l'engrais. Les règles de décision fournies par la stratégie sont du type : *le 16 mars, fertiliser les parcelles 3, 4, 6 et 7*. Une généralisation a été proposée afin d'obtenir des règles plus souples (surtout au niveau des dates) sur des catégories d'exploitation agricoles similaires décrites par des descripteurs [LZC16]. Les règles de décision proposées sont alors du type :

— Fertiliser la prairie :
— si la date est entre le 15 Mars et le 1^{er} Avril, et
— la distance entre la prairie et le bâtiment de traite est entre 400 et 800 mètres, et
— la surface de la prairie est supérieure à 2.5 ha.

3.7 Intégration de facteurs socio-économiques

La recherche d'une stratégie optimale nécessite de définir les coûts qui relèvent généralement des critères économiques ou sociaux autour des écosystèmes. Dans la dernière partie de ce chapitre, nous proposons une extension d'EcoMata qui modélise des stratégies de management et se place au dessus de l'unité AP en

contrôlant les automates décrivant les activités humaines [LBZ16].

Stratégies de management
Nous avons choisi de représenter les stratégies de management à l'aide d'un système de règles de production. Ces règles permettent dans cette première approche une représentation aisée permettant de tester l'intérêt des stratégies. L'approche est illustrée sur l'écosystème marin d'Ouvéa pour lequel nous disposons déjà du modèle et sur lequel nous pouvons collaborer avec nos partenaires écologues. Les règles de management énoncées par les experts sont les suivantes :

- Les espèces trophiques de haut-niveau sont celles qui proposent la plus grande valeur marchande.
- Les espèces de haut-niveau sont exploitées prioritairement.
- On bascule la pêche vers un niveau inférieur dès que la biomasse des niveaux trophiques supérieurs atteignent un seuil limite vis-à-vis de leur résilience.
- On bascule de nouveau vers le niveau trophique supérieur selon deux options :
  - *Scénario 1 : changement minimal.* Une espèce est pêchée tant qu'elle n'a pas atteint le seuil limite.
  - *Scénario 2 : préférence au prédateur.* La pêche des espèces de haut-niveau est toujours privilégiée à partir du moment où celles-ci disposent des ressources suffisantes.

Ces stratégies vont guider les politiques d'exploitation et donc définir de manière adaptative les automates de pêche. La stratégie adaptative a pour conséquence un contrôle des pêches par le niveau de biomasse des espèces. Conceptuellement, cela signifie que les événements envoyés par les espèces lors d'un changement d'état sont pris en compte par les automates pêches. Ces automates ne sont plus construits à partir des chronogrammes donnés par les utilisateurs mais à partir des règles de production des stratégies. Leur évolution est guidée à la fois par le niveau de biomasse des espèces et par les stratégies de management. Cette extension n'a pas d'impact sur la complexité du modèle global. Chaque nouvelle stratégie nécessite la reconstruction du modèle ce qui n'est pas problématique.

Une illustration de la représentation des stratégies par des règles est donnée dans le tableau ci-dessous avec deux forces de pêches considérées : f_H pour une pêche intensive et $f_stopped$ pour l'arrêt de la pêche. Le prédateur est $sp5$ et sa pression de pêche $f5$, la proie est $sp4$ et sa pression de pêche $f4$.

Scénario 1 : changement minimal	Scénario 2 : préférence au prédateurs
init : $f4_stopped, f5_stopped, sp4_N, sp5_N$	init : $f4_stopped, f5_stopped, sp4_N, sp5_N$
$(sp5_N \vee sp5_H) \wedge f4_stopped \implies f5_H!$	$(sp5_N \vee sp5_H) \implies f5_H! \wedge f4_stopped!$
$sp5_L \implies f5_stopped!$	$sp5_L \implies f5_stopped!$
$(sp4_N \vee sp4_H) \wedge f5_stopped \implies f4_H!$	$(sp4_N \vee sp4_H) \wedge f5_stopped \implies f4_H!$
$sp4_L \implies f4_stopped!$	$sp4_L \implies f4_stopped!$

Les coûts sont donnés par les écologues : la valeur marchande des espèces, le coût de la pêche (flotte, engins). L'intérêt de l'approche est de pouvoir vérifier le modèle sur un horizon à long terme. Nous avons étudié les deux scénarios sur 30 ans et proposé une synthèse des trajectoires. Le modèle des stratégies va nous aider à définir les chroniques de pêches à proposer aux gestionnaires. Les stratégies génèrent des comportements génériques qui sont instanciés temporellement par la dynamique du modèle en particulier les contraintes d'arrêt de la pêche si l'on atteint un état seuil pour les espèces. L'usage du patron de requête $WhichStates$ sur différentes dates d'intérêt nous permet de récupérer les trajectoires de l'écosystème et de découvrir les chroniques qu'il faudrait envisager pour la pêche.

Nous avons testé cette approche sur le lagon d'Ouvéa pour lequel nous avions déjà un modèle et des connaissances socio-économiques. Deux pressions de pêche, une sur le prédateur et une sur une des proies ont été définies. Les trajectoires de l'exploration du modèle pour le scénario 2 (la pêche du prédateur est favorisée) sont présentées figure 3.8.



FIGURE 3.8 – Chroniques de pêche adaptative sur le modèle de l'écosystème d'Ouvéa. Les bandes verticales rouges correspondent à des périodes sans pêche possible.

Même si nous nous sommes placés ici dans un cas assez excessif avec une pêche forte, nous pouvons remarquer qu'une exploitation commerciale de cette zone marine serait compliquée. Nous constatons que seules de longues périodes sans activité de pêche permettent au stock de récupérer. Pour assurer la durabilité du lagon, la pratique de la pêche ne pourrait s'appliquer que sur la moitié du temps par périodes alternatives entre le prédateur et la proie. Contrairement aux idées reçues, ce n'est pas en privilégiant l'espèce la plus rentable que le profit est le meilleur mais le scénario qui préfère la proie tant que possible.

3.8 Conclusion

Dans ce chapitre nous avons proposé des méthodes variées pour répondre aux questions d'aide à la décision pour la gestion les écosystèmes. Notre point de vue a été de proposer des modèles *holistiques*, qui nous ont menés à considérer des représentations volontairement abstraites et peu paramétrées permettant de répondre de manière plus synthétique à des problèmes complexes. Les diverses études sur des applications réelles confirment le choix du formalisme à événements discrets et en particulier des automates temporisés qui semble très adaptés pour représenter des entités en interaction et où la dynamique du système dépend fortement des interactions mutuelles entre les entités.

Les approches proposées sont relativement génériques et pourraient s'appliquer à tout écosystème composé d'un ensemble d'entités ayant une dynamique temporelle et interagissant entre eux. La modélisation utilisée permet de la représentation de systèmes mal connus, *non déterministes* avec des *incertitudes sur les contraintes temporelles* décrivant la dynamique du système. Pour utiliser la synthèse de stratégies, nous avons pu adjoindre au modèle des *coûts* et des notions de *contrôlabilité* sur les événements. Le non-déterminisme a cependant du être levé dans une certaine mesure pour la recherche de la stratégie optimale. La modélisation qualitative qui était une option cruciale au démarrage de cette étude a été adaptée pour l'outil PaturMata qui utilise un modèle hybride.

Cependant aujourd'hui même les écosystèmes naturels sont de plus en plus *connectés*. En quelques années, des capteurs numériques sont apparus dans les champs, les cours d'eau, les étables et même sur les animaux mettant à disposition des données produites quotidiennement. C'est dans cette perspective que se dessinent les orientations de ce travail : introduire des méthodes de machine learning pour améliorer l'apprentissage des modèles et les tâches d'aide à la décision.

Chapitre 4

Fouille de données et Aide à la décision

Sommaire

4.1 Contexte Scientifique 89
4.2 Détection des anomalies à partir d'un flux de données 90
4.3 Machine-Learning en Agro-écologie 94
4.3.1 Tendances actuelles pour la gestion des agro-écosystèmes 95
4.3.2 Travaux en cours : prédiction de la ration alimentaire des truies allaitantes 99
4.4 Comprendre ses données pour mieux les traiter 102
4.4.1 Test d'unimodalité 103
4.4.2 Recherche du k dans la méthode des k plus proches voisins 105

4.1 Contexte Scientifique

Ces dernières années, les masses de données croissantes constituent un véritable potentiel pour extraire des connaissances exploitables permettant de fournir des services d'aide à la décision de plus grande qualité. Le défi scientifique relève de l'intégration des approches de fouille, de représentation des connaissances et de raisonnement pour fournir des méthodes interactives pour des non-spécialistes leur permettant de s'assurer de prendre la meilleure décision. Si l'on se reporte aux phases du processus d'aide à la décision présentée chapitre 1, *représentation et formalisation du problème*, *modèle d'évaluation*, *recommandation finale*, une récente

étude montre que les approches à l'aide de la fouille de données se concentrent largement sur les deux premières étapes [MMH15]. Le processus de choix est souvent considéré comme une phase "post-process" où le décideur doit intégrer lui-même les résultats de la fouille de données dans son processus de décision.

Ce contexte a coincidé avec l'évolution de mon équipe de recherche IRISA/INRIA passée de DREAM (Diagnostic et Recommandation d'Actions) à LACODAM (Large scale Collaborative Data Mining) en 2015 avec pour nouvelle thématique la fouille de données et l'Intelligence Artificielle pour l'explicitation du sens ou l'aide à la décision. Les travaux actuels m'ont amené à m'intéresser à plusieurs thématiques relevant de ce domaine.

On peut tout d'abord citer les contributions d'Alban Siffer, doctorant que je co-encadre avec Alexandre Termier et Pierre-Alain Fouque de l'IRISA dans le cadre d'une thèse Cifre avec la société Amosys (spécialisée en cyber-sécurité). Il travaille sur la détection d'anomalies à partir de séries temporelles, permettant d'identifier un comportement inattendu comme, par exemple, un pic de contamination de l'eau, des événements climatiques exceptionnels, mais aussi les intrusions sur les réseaux informatiques. Pour sa part, dans le domaine des sciences animales, Raphaël Gauthier se concentre sur un problème d'apprentissage sur des séries temporelles dans le but de proposer la ration alimentaire la plus adaptée des truies allaitantes. Ce doctorant fait sa thèse dans le cadre de l'institut de convergence #DigitAg dédié à l'agriculture numérique. Je le co-encadre avec Jean-Yves Dourmad, agronome à l'INRA de St-Gilles (35). Je présenterai ses travaux après une synthèse sur les techniques de fouille de données aujourd'hui employées dans le domaine de l'agro-écologie. Enfin, je me suis intéressée à une partie un peu plus éloignée des problèmes d'aide à la décision des utilisateurs avec la deuxième partie du travail d'Alban Siffer qui relève plus de la compréhension des données en amont de la fouille des données. Les concours de circonstances et les profils d'étudiants nous portent parfois vers de nouvelles voies avec cette partie plus orientée vers la science des données.

4.2 Détection des anomalies à partir d'un flux de données

De nombreuses applications doivent aujourd'hui gérer des données datées générées en continu et à grande vitesse. La détection des anomalies dans les séries temporelles est un sujet crucial pour des domaines tels que la détection des intrusions, la consommation énergétique, la finance mais également en écologie pour la détection d'événement exceptionnel sur des données climatiques ou hydrologiques.

Le problème de la détection des anomalies consiste à identifier des données, des observations ou des événements appelés anomalies (*outlier* en anglais) qui ne sont pas conformes a ce qui peut-être attendu dans un groupe de données. La définition qui nous parait le plus précise est celle donnée par [CBK09] : "Les instances de données normales se produisent dans les régions à probabilité élevée d'un modèle stochastique, tandis que les anomalies se produisent dans les régions à faible probabilité du modèle stochastique". Ce champ d'investigation est largement couvert dans la littérature pour les données statiques et repose sur le calcul de distance à l'aide de méthodes telles que les plus proches voisins ou d'autres techniques de clustering. La principale contrainte des flux de données relève de l'incapacité de visiter plusieurs fois les données permettant de garantir une faible complexité pour autoriser un traitement en ligne, tout en tenant compte de l'évolution possible des références ou concepts au cours du temps. Une première catégorie d'algorithmes proposent de s'appuyer sur le calcul de distance pour la détection des anomalies (STORM [ABP06], CORM [ELN+08], DBOD-DS [SG10]). Ces méthodes nécessitent cependant l'intervention de l'utilisateur pour définir les seuils limitant leur usage pratique. Des méthodes statistiques s'affranchissent de cette contrainte mais s'appuient sur une hypothèse qui est la distribution des données, information nécessaire à l'apprentissage des modèles. [LF14] propose la détermination d'un seuil probabiliste permettant de discriminer les données normales et anormales. La limite de cette méthode se trouve dans le nombre important mais nécessaire d'échantillons d'apprentissage permettant l'obtention d'un taux faible de faux positifs.

Les contributions d'Alban Siffer proposent une méthode de détection des anomalies pour des séries temporelles univariées et unimodales sans aucune hypothèse de distribution a priori sur les données. L'approche repose sur la théorie des valeurs extrêmes [BGST06]. Cette théorie met en lumière le résultat suivant : la distribution des valeurs extrêmes est quasi indépendante de la distribution des données. La théorie des valeurs extrêmes peut ainsi inférer la distribution des événements extrêmes sans hypothèse forte sur la distribution originale, ce qui permet de prédire des valeurs extrêmes sans précédent (comme des crues exceptionnelles ou des vagues anormalement hautes par exemple).

Le problème fondamental peut être exprimé de la manière suivante :

Soit X une mesure d'interêt et $X_1, \dots X_n$, n des observations indépendantes de cette mesure. Est-il possible de trouver un seuil z_q telle que la probabilité de voir $X > z_q$ est plus petite que q (avec q aussi petit que souhaité) ?

L'objectif de la théorie des valeurs extrêmes consiste à définir la loi de distribution de ces valeurs extrêmes. D'après [FT28] et [Gne43], les événements extrêmes suivent une même distribution appelée Distribution des Valeurs Extrêmes (DVE) indépendamment de leur distribution d'origine. Pour la plupart des distributions, la probabilité décroit lorsque les valeurs sont extrêmes : $\mathbb{P}(X > x) \to 0$ quand x croit. La fonction $\bar{F}(x) = \mathbb{P}(X > x)$ représente la queue de la distribution de X. La DVE a pour objectif de définir la distribution de cette queue. En définissant une DVE de la queue d'une distribution en entrée inconnue, il est donc possible d'évaluer la probabilité des valeurs extrêmes. La contribution s'appuie sur le théorème POT (*Peaks-Over-Threshold*) [BDH74, PI75] en calculant l'estimateur grâce à la méthode du maximum de vraisemblance qui a le mérite d'être efficace et robuste. Il est ainsi possible d'estimer z_q tel que $\mathbb{P}(X > z_q) < q$ sans aucune hypothèse sur la distribution de X et sans même disposer de connaissance sur sa distribution. L'idée repose sur une phase d'initialisation (ou calibration) qui fixe un seuil relativement haut t à partir des n premières observations, recherche les valeurs extrêmes (supérieures à ce seuil) et définit une loi de Pareto Généralisée (GPD) qui les représente. Les détails théoriques de l'approche proposée se trouvent dans la publication suivante [SFTL17].

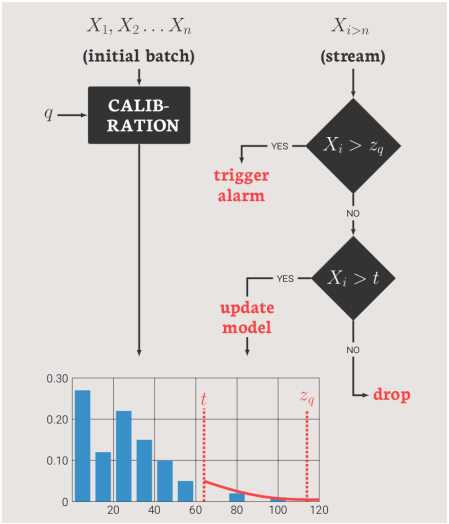


FIGURE 4.1 – Principe de l'algorithme SPOT

À partir de ce résultat, il est possible d'estimer z_q et d'envisager un algorithme de détection des anomalies sur une série temporelle. Une première phase d'initialisation calcule un seuil z_q à partir de n observations $X_1, \dots X_n$. Ce seuil est ensuite mis à jour à partir des données entrantes et est utilisé comme critère de décision pour la détection des anomalies. Deux algorithmes ont été proposés : SPOT qui

travaille sur des cas stationnaires et DSPOT (pour Drift SPOT) qui prend en considération le changement de concept. L'algorithme SPOT est capable de distinguer les valeurs extrêmes, les valeurs supérieures au seuil z_q, des anomalies, les valeurs supérieures au seuil t qui sont alors utilisées pour mettre à jour le modèle (cf. figure 4.1).

Pour le cas de DPSOT, on considère que la distribution des valeurs peut évoluer au cours du temps comme la saisonnalité par exemple. Dans ce cas, les valeurs relatives entre variables, au sein d'une fenêtre temporelle, sont également prises en compte dans l'algorithme. Afin d'optimiser la vitesse et la robustesse de l'algorithme des optimisations ont été proposées pour l'estimation des paramètres, en particulier l'usage d'une méthode analytique pour la recherche des racines de la fonction de vraisemblance (cf. [SFTL17] pour plus de détail).

Des expérimentations ont été menées sur des jeux de données réels, disponibles sur internet, dans plusieurs domaines : réseaux, physique et finance. La fiabilité de la méthode a été évaluée en comparant le taux d'erreur de l'algorithme utilisant le seuil z_q estimé par rapport au seuil théorique en fonction des observations et du nombre d'observations n prises en compte pour l'initialisation. On constate que le seuil converge quel que soit n, la seule contrainte étant de prendre un n suffisamment grand pour être capable d'évaluer correctement la DVE. Une des expérimentation porte sur le cours de la bourse des actions EDF au sein d'une journée. Le 9 février 2017, une explosion a eu lieu sur le site de Flamanville, à 11 heures. Nous avons appliqué DSPOT sur les données du cours pour cette période pour laquelle nous disposons d'une donnée par minute. Sur la figure 4.2, on constate que DSPOT suit bien le comportement des données et une anomalie est détectée aux alentours de 11 heures. L'algorithme a été exécuté avec $q = 10^{-3}$ et la taille de la fenêtre fixée à 10 observations. Les tests de performances ont montré que ces algorithmes étaient capables d'analyser des séries temporelles en traitant plus de 1000 valeurs par seconde.

Ce travail permet de proposer un outil efficace, sûr et simple d'usage pour la détection des anomalies sur les séries temporelles. Son principal intérêt est qu'il peut traiter tout type de données sans aucune connaissance sur leur distribution initiale, sans avoir à fixer manuellement un seuil et qui s'adapte à l'évolution des données dans le temps. Alban Siffer a réalisé une bibliothèque Python [1] déjà utilisée par les membres de notre équipe pour analyser leurs données qui y trouvent un réel bénéfice. C'est le premier travail à proposer l'approche de la théorie des valeurs extrêmes pour la recherche des anomalies sur les séries temporelles. Sa

1. Les algorithmes SPOT et DSPOT sont disponibles ici : https ://github.com/Amossys-team/SPOT

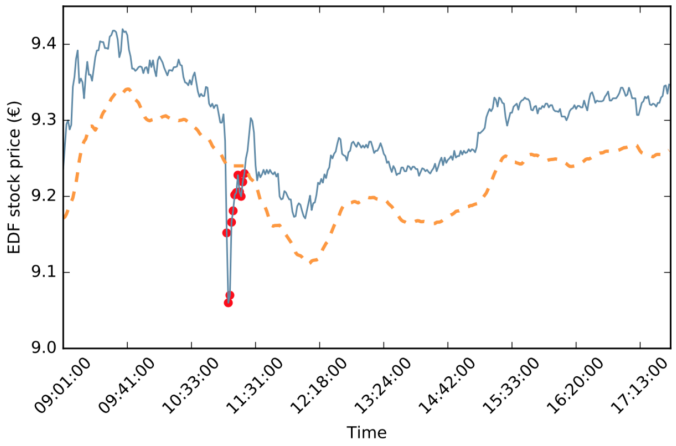


FIGURE 4.2 – Exécution de l'algorithme DSPOT sur le prix des actions EDF le 9 février 2017 (explosion sur le site de Flamanville)

limitation est qu'aujourd'hui l'approche ne traite que le cas des séries univariées. La perspective la plus évidente serait sans doute de s'intéresser au problème multivarié, toujours avec cette même théorie qui offre de nombreux pans non explorés, mais difficiles d'accès.

4.3 Machine-Learning en Agro-écologie

Je me suis également intéressée à l'usage de la fouille de données dans le domaine de l'agro-écologie. Si ce domaine a mis plus de temps à être touché par la vague des données, il a rapidement rattrapé son retard. On parle maintenant des *fermes connectées* ou des *océans numériques* et l'agriculture de précision consiste à analyser au plus près les structures des sols afin de proposer les quantités de produit nécessaires à une production optimale. Cette évolution scientifique a nécessairement pour conséquence d'intégrer les méthodes de data-mining ou de machine-learning ce qui est relativement nouveau dans le domaine.

Dans cette section, je souhaitais faire un point sur les techniques de data-mining et de machine-learning appliquées à l'agriculture ou à la gestion des systèmes environnementaux afin d'analyser la particularité du domaine. Dans un deuxième temps, je présenterai les travaux menés pour l'alimentation de précision des truies en lactation dans le cadre de la thèse de Raphael Gauthier.

4.3.1 Tendances actuelles pour la gestion des agro-écosystèmes

La particularité du domaine se caractérise surtout par la nature des données :

— *hétérogénéité des données.* Les systèmes agro-environnementaux sont aujourd'hui observés par une abondance de données : les données numériques issues de capteurs, collectées auprès des institutions, mais également les expertises humaines, les données qualitatives comme les observations des agriculteurs à l'aide d'un smartphone voire les retours collectés auprès des réseaux sociaux (apparition d'événements climatiques, maladies constatées dans les exploitations, etc.) [MBK+19a].

— *Incertitudes des données.* Les données sont sujettes à une forte incertitude liée à un milieu non protégé (capteurs en plein air ou sur des animaux, par exemple) avec pour conséquence des données incorrectes ou manquantes soumises au bruit et aux anomalies. L'idéal serait donc de disposer de nombreuses données permettant la correction de ces imperfections.

— *Variabilité temporelle.* Les données sont issues de capteurs qui enregistrent les informations temporelles à des échelles de temps variables (de la seconde à la semaine).

— *Variabilité spatiale.* C'est surtout le cas en écologie où il est nécessaire de comprendre la distribution spatiale des processus. Les capteurs sont positionnés selon un plan d'échantillonnage avec un découpage de la zone généralement en deux ou trois dimensions spatiales.

— *Données multivariées.* Les données issues de plusieurs capteurs doivent être analysées simultanément afin de garantir l'analyse correcte d'un système complexe régi par un grand nombre de variables.

On pourrait reformuler ces caractéristiques selon les 3 "V" du *big-data* (Volume, Velocity, Variety) bien que le volume des données tout comme leur vélocité soient aujourd'hui encore un peu justes pour considérer les données agro-environnementales comme du véritable *big-data*. Une exception relève des applications utilisant des images de télédétection ou de drones qui, elles, manipulent des données de grande taille. Leur variété est certainement ce qui les caractérise le plus. Avec l'évolution rapide des capteurs et des mentalités, le *big-data* devrait cependant arriver sous peu surtout pour le domaine de l'agriculture (par rapport à l'écologie) étant donné les enjeux économiques.

Traditionnellement, ces données ont été traitées par des méthodes d'analyse statistique qui aujourd'hui se trouvent confrontées à la complexité des jeux de données. Les approches de data-mining utilisées avec succès dans d'autres domaines investissent donc la sphère de l'agro-écologie depuis peu, et les publications sur le sujet explosent depuis 2015-2016 [MNA17].

L'exploration de données ou data-mining offre la capacité de fournir, sur un volume important de données, des informations significatives et utiles à l'aide de processus automatiques ou, plus souvent, semi-automatiques. L'idée consiste à proposer des régularités, encore appelées *patterns*, intéressantes pour la compréhension des données et pour éventuellement faire des prédictions sur des données futures. Pour sa part, l'apprentissage automatique ou machine-learning offre aux ordinateurs la capacité d'apprendre un modèle à partir des observations afin de pouvoir l'utiliser sur un problème de même nature. Le machine-learning repose sur les techniques de base du data-mining puisque de nombreuses approches recherchent les descriptions structurelles entre les données avant de les employer sur de nouveaux exemples [HKP12, WFHC17].

Alors que les publications sur la fouille de données des agro-écosystèmes étaient très rares encore récemment, on dénombre sur *Sciences Direct* plus de 2000 articles pour l'année 2018. La caractéristique de ces publications est qu'elles portent sur les problèmes applicatifs très spécifiques dans des revues du domaine. Il est assez difficile d'avoir une vision relativement complète des travaux menés du point de vue de l'IA. Des articles de synthèse sont cependant disponibles, que l'on peut classer de la manière suivante :

— **big data.** [LKJ+16, JPM+17] explorent les enjeux et les potentialités des données pour une nouvelle génération de systèmes en agriculture. [KKPB17] recense les données, méthodes et outils utilisés pour différents types de problèmes en agriculture (productions animales, cultures, sols, adventices, biodiversité, etc.). [MBK+19b] s'intéresse plus particulièrement à l'extraction des connaissances à partir des données issues des objets connectés IoT mis à la disposition des agriculteurs.
— **data-mining.** [PK17, KR16] sont des bibliographies similaires qui proposent de détailler les techniques de data-mining employées en agriculture récemment. Des problèmes de classification de cultures ou des sols sont traités par les méthodes classiquement mises à disposition dans les bibliothèques des langages. Pour sa part, [MNA17] compare des méthodes de clustering (PAM, CLARA, DBSCAN) pour définir des systèmes de production de cultures similaires à partir de données variées (climatiques, pédologiques). Le choix de la méthode est abordé dans [GISM+18] qui s'appuie en premier lieu sur la présence ou non, dans le jeu de données, de variables explicatives et présente les méthodes employées dans les sciences environnementales (qualité des eaux, de l'air, description de paysages, des espèces, identification climatiques : cyclones, anomalies de températures).
— **machine-learning.** Il apparait dans [LBM+18] que la plupart des articles portent sur les prédiction de cultures et que les méthodes les plus employées s'appuient sur les réseaux de neurones. Le sujet de la protection

des plantes est plus particulièrement présenté dans [IAS+18] s'intéressant à l'apprentissage de modèles graphiques probabilistes offrant une meilleure interprétabilité. En ce qui concerne les services écosystémiques, [HWP+18] présente l'intérêt du machine-learning en particulier pour la gestion des sites gérés par une modélisation difficile à paramétrer.

— **deep-learning.** [KPB18a] se concentre sur le deep-learning en agriculture en recensant une quarantaine de travaux consacrés à ce domaine et publiés après 2015. Ils concernent principalement des applications s'appuyant sur un grand volume de données comme des images ou du texte issus d'entrepôts de données avec pour champ d'application la prédiction de cultures ou des mauvaises herbes, ainsi que le comptage de fruits.

Je propose ici un récapitulatif des approches référencées dans les différents articles de synthèse (cf. figure 4.3). Il décrit les méthodes de machine-learning employées en fonction des champs d'application de l'agro-écologie.

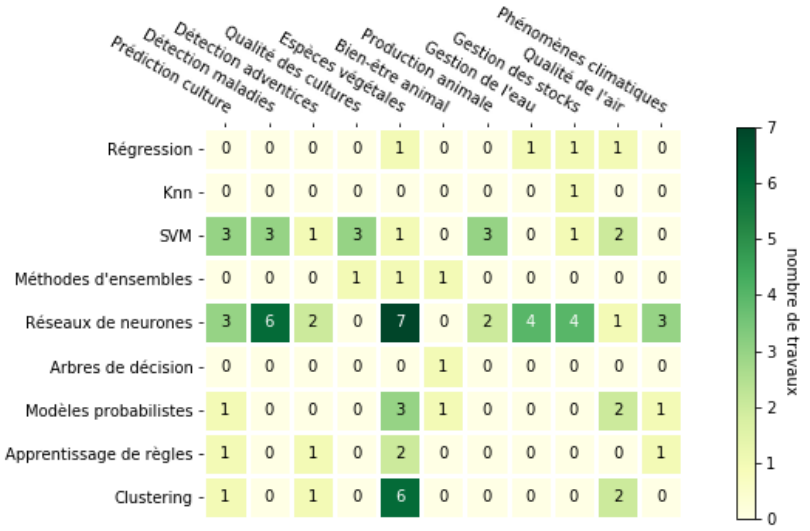


FIGURE 4.3 – Les méthodes de machine-learning employées dans les domaines de l'agro-écologie

Si l'on s'intéresse au point de vue de l'aide à la décision, les types de problèmes traités sont relativement similaires :

— **Comptage.** Par exemple le comptage des grains de café sur les branches à partir d'images, ou le comptage des fruits récoltés.

— **Détection.** La détection des maladies ou des insectes sur les cultures à partir d'images comme la détection précoce des problèmes de production des œufs de poules à partir de données enregistrées sur l'exploitation.

Ce problème peut s'apparenter à la classification encore appelé reconnaissance puisqu'il s'agit de partitionner les données selon des catégories (malade/sain, espèces végétales, qualités des cultures).

— **Prédiction.** La prédiction du stade de développement d'une culture ou de sa récolte à partir d'images, la prédiction des acides gras volatils chez les ruminants à partir de données expérimentales.

— **Supervision.** Peu de travaux portent encore sur le sujet, mais les récents travaux sur le suivi du comportement des animaux pour évaluer leur santé et leur bien-être [MMPK17, SCL+19] sont encourageants.

On constate que la majorité des articles est consacrée à la prédiction des cultures et à la détection des maladies où la disponibilité des images est suffisante pour l'application de techniques de fouille de données. Il s'agit d'un terrain de jeu parfait pour l'application du *deep-learning* avec des problèmes classiques de classification ou de prédiction, et de nombreuses images disponibles. Ce contexte explique l'accroissement de publications dans ce sens depuis 2017. Dans les articles offrant des comparaisons avec d'autres techniques, le deep-learning a montré son intérêt en terme de résultats. Bien que la majorité des publications utilisent des architectures à base de réseaux neuronaux convolutifs (CNN) [KPB18b] on trouve également l'emploi de réseaux de neurones récurrents pour l'analyse de séries temporelles [RK17]. Un inconvénient majeur néanmoins des réseaux neuronaux, mais cela vaut également pour les autres méthodes de machine-learning mais dans une moindre mesure, est la nécessité de disposer d'un volume très important de données d'entrainement et l'annotation des données doit être effectuée par des experts du domaine parfois difficiles à trouver [ABA17]. Enfin, le principal reproche fait aux techniques de machine-learning est qu'elles peuvent apparaître comme des boites noires [OPK+17] qui fournissent des prédictions potentiellement explicables mais d'une manière que les hommes ne peuvent les comprendre. Les décideurs ont alors avoir du mal à leur accorder leur confiance. J'aborderais ce point dans les perspectives en présentant l'*interprétabilité* des modèles de machine learning. Les réponses apportées par les techniques de data-mining en agro-écologie sont aujourd'hui fiables mais se limitent à des problématiques simples (d'un point de vue conceptuel et non technique) : classer, reconnaitre, prédire. Les problèmes de planification, de jeux de scénarios, de recherche de stratégies pourraient être extrapolés des modèles appris. Enfin, les caractéristiques des données agro-environnementales présentées plus haut sont rarement évoquées dans le choix des méthodes et sont sans doute traitées manuellement lors des pré-traitements nécessaires à l'emploi des méthodes traditionnelles de machine-learning.

4.3.2 Travaux en cours : prédiction de la ration alimentaire des truies allaitantes

Comme de nombreuses approches évoquées précédemment, les contributions de cette section s'inscrivent dans l'objectif d'apporter une meilleure prédiction à l'agriculteur. Ce travail est mené dans la thèse de Raphaël Gauthier en collaboration avec l'INRA dans le cadre de l'institut de convergence #DigitAg. Il s'intéresse à la prédiction de la ration alimentaire des truies en lactation pour l'élevage de précision.

Contexte.
L'amélioration des systèmes d'alimentation constitue une voie de réduction de l'utilisation des ressources non renouvelables et de prévention des rejets de polluants dans l'environnement. La maîtrise de l'alimentation permet également d'adapter l'élevage aux multiples attentes de la société et donc de limiter la compétition entre alimentation humaine et animale pour l'utilisation des mêmes ressources végétales. Enfin, le respect, la santé et le bien-être animal passe par une alimentation adaptée. En raison de la variabilité de la production laitière et de l'appétit des truies, les besoins nutritionnels sont très différents d'une exploitation à l'autre ou d'un lot de mise bas à l'autre. Aujourd'hui, dans la pratique, le même régime est généralement appliqué à toutes les truies d'un troupeau.

De nouveaux capteurs permettent d'acquérir des données en élevage et de manière individualisée lorsque l'animal est muni d'une boucle de type RFID. D'autres données sont également possibles comme l'environnement de chaque animal (température, humidité, etc.), ainsi que des données historiques relatives à l'animal lui-même (ex : lactation précédente), à des animaux similaires de la même ferme ou d'exploitations similaires. L'intégration de modèles de prédiction dans les outils d'aide à la décision qui pourraient éventuellement être intégrés aux équipements d'alimentation est donc un enjeu important pour les éleveurs de porcs. La question de recherche porte sur la détermination en temps réel des besoins nutritionnels des truies allaitantes et sa mise en pratique : "Comment pouvons-nous décider en temps réel de la composition optimale et de la quantité d'aliments à donner aux truies allaitantes en utilisant toutes les informations disponibles à ce moment ?"

Problématique.
On cherche à déterminer la satisfaction des besoins naturels d'une truie allaitante pour un jour j. L'échelle de temps considéré est : $1, ..., \tau$, avec τ le jour du sevrage et le jour pour pas de temps. Sbn_j définit la satisfaction des besoins naturels en fonction de $Alim_j$, la composition de la ration, et de Ing_j l'ingestion de la truie.

La composition de la ration s'appuie essentiellement sur les besoins nutritionnels (énergie métabolisante, lysine digestible, phosphore digestible) et définit les quantités de ces nutriments nécessaires pour couvrir les besoins d'entretien et de production laitière. La caractérisation de l'ingestion permet de définir l'appétit de la truie qui dépend du jour dans la période de lactation mais également de la nature de l'animal. Par la suite, il s'agit de proposer la composition optimale d'aliments permettant de satisfaire ses besoins en rapport avec ses facultés d'ingestion.

La satisfaction des besoins de la truie au jour j est définie par :
$Sbn_j = f(Alim_j, Ing_j)$ tels que :

- $Alim_j$ est l'aliment au jour j avec $Alim_j = f(Bes_j, min(C), min(R))$ avec Bes_j les besoins nutritionnels par jour de chaque truie, C le coût des aliments, et R les rejets des nutriments.
- Ing_j sa capacité d'ingestion prédite au jour j avec $Ing_j = f(X, [Ing'_1, \cdots, Ing'_{j-1}])$ avec X un ensemble de séries temporelles de consommation sur la période de lactation, et $[Ing'_1, \cdots, Ing'_{j-1}]$ l'historique de la consommation réelle de la truie.

La définition de l'aliment $Alim_j$ permet de définir les proportions permettant d'obtenir un mélange d'aliments (un aliment A riche en nutriments, et un aliment B plus pauvre et moins cher). L'ingestion au jour j, Ing_j est déterminée à partir d'un ensemble de séries temporelles X, qui donne l'ingestion au jour le jour d'un ensemble de truies précédemment observées, et de l'historique de la consommation réelle de la truie depuis le début de la lactation. Cette consommation réelle, Ing'_i, peut être déterminée à partir d'un capteur qui mesure ce qui n'a pas été consommé.

Démarche.
L'originalité de la démarche consiste à développer sur un système d'aide à la décision qui s'appuie sur la combinaison de deux méthodes (cf Figure 4.4).

- Un modèle mécaniste centré sur l'individu permettant de définir les besoins nutritionnels par jour Bes_j. Ce modèle s'appuie sur la production laitière qui est estimée à partir de données disponibles sur la truie.
- Un module de prédiction de l'ingestion s'appuyant sur les séries temporelles.

La première partie de cette thèse, démarrée en novembre 2017, a été consacrée au développement du modèle mécaniste. C'est une pièce indispensable à l'architecture globale qui présente un grand intérêt pour nos partenaires de l'INRA puisque c'est le premier modèle qui repose sur la représentation individuelle des truies en lactation pour prédire leur besoin alimentaire. Les besoins nutritionnels de la truie sont estimés à partir des travaux de Noblet et Etienne [NE89] qui ont directement

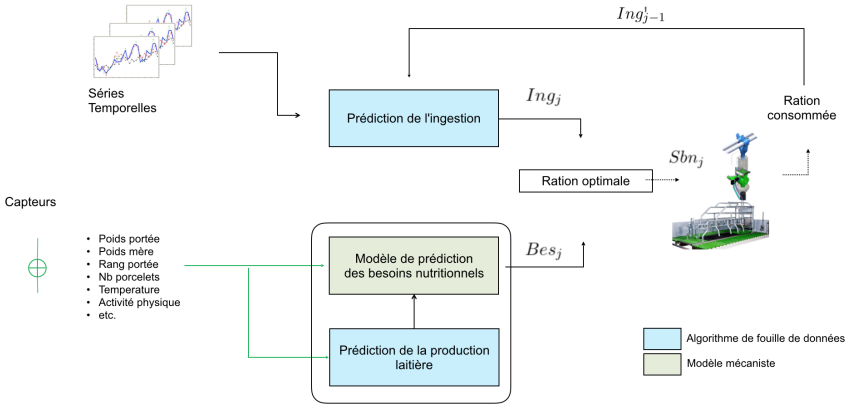


FIGURE 4.4 – Démarche générale de l'outil d'aide à la décision pour l'alimentation des truies allaitantes

relié les nutriments exportés dans le lait à la croissance moyenne et à la taille moyenne de la portée. Le modèle a été développé en Python et permettra par la suite l'intégration des modules de fouille de données. Il est décrit en détail dans la publication suivante [GLG+19].

La seconde partie de la thèse qui démarre actuellement concerne les aspects fouille de données permettant d'estimer la production laitière et l'ingestion de la truie. L'estimation de la production laitière est indispensable au modèle mécaniste pour fournir les besoins nutritionnels les plus précis possibles pour chaque truie. Ce modèle repose aujourd'hui sur le travail de méta-analyse de Hansen et al. [HSK+12] qui reprend une courbe largement utilisée pour la production laitière des vaches en adaptant les paramètres à la production laitière des truies. Des prédictions à partir de données expérimentales permettraient sans doute des estimations plus réalistes. Nous disposons de données récoltées auprès du centre de développement du porc du Québec, provenant de six élevages, qui totalisent plus de 100 000 lactations. Pour chaque animal, trente paramètres sont disponibles caractérisant la truie elle-même, et la portée. Le plus souvent, les porcelets ne reçoivent pas d'alimentation solide avant le sevrage. Par conséquent, la croissance de la portée est principalement due à la consommation de lait, ce qui fait du gain de poids de portée au sevrage un bon indicateur de la production laitière des truies. Cependant, la prédiction du poids de portée au sevrage est rendue difficile par de multiples sources de variations, inhérentes à la truie elle-même, ses conditions de l'élevage et son environnement. Des analyses statistiques ont déjà été mises en œuvre avec succès [NQH+12] mais l'utilisation d'algorithmes de fouille de données permettrait sans doute des prédictions plus précises. À partir des nombreuses données

disponibles, les recherches s'orientent vers l'emploi de méthodes d'apprentissage supervisées pour prédire le poids de la portée au sevrage.

Pour l'ingestion, il s'agit d'apprendre un modèle de comportement des classes de truies à partir des séries temporelles par des méthodes non supervisées. La consommation d'aliments d'une truie i au cours de la lactation peut être vue comme une série temporelle X^i univariée, telle que $X^i = [X_1, X_2, \cdots, X_n]$ avec X_1 la consommation au jour 1 et n la durée de la lactation. De nombreux autres paramètres peuvent influencer la consommation de la truie comme ses facteurs propres, sa production laitière, son environnement et d'éventuelles perturbations sanitaires. Aucun modèle mécaniste ne peut prédire l'ingestion des truies et il s'agit alors de s'appuyer sur des données existantes. L'analyse des données par les agronomes a mis en lumière des phases de comportements typiques que nous envisageons d'identifier à l'aide de la reconnaissance de *shapelets* permettant d'inférer ces états. Nous pourrons alors définir un modèle par classe d'animaux. Ces modèles seront ensuite utilisés en temps réel pour prédire l'ingestion et superviser le comportement des truies, en permettant le recalage c'est-à-dire le changement de modèle pour un individu si leur comportement était trop déviant. L'intérêt des *shapelets* repose essentiellement sur l'interprétabilité des résultats et leur efficacité [KR13]. Nous pensons utiliser une approche qui combine les techniques de classication par intervalles et par shapelets. La classification par intervalles permet de découper la série en sous-séries pouvant être représentées de manière symboliques au sein desquelles des shapelets pourront être identifiés [LNGII19].

4.4 Comprendre ses données pour mieux les traiter

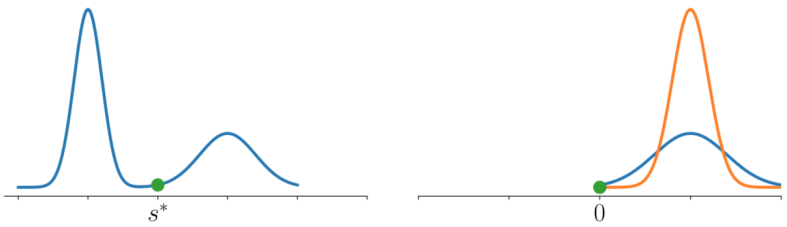
La dernière partie de ce chapitre présente les récents travaux d'Alban Siffer sur la compréhension des données avec pour objectif l'emploi des techniques d'apprentissage les plus appropriées. Cette section s'éloigne de l'aide à la décision, en tout cas d'un utilisateur non-informaticien, c'est la raison pour laquelle nous aborderons brièvement ces approches. Elles présentent cependant des résultats très intéressants pour le domaine de la science de données en proposant des outils utiles avant tout traitement de fouille. Nous présentons dans un premier temps le test d'unimodalité avant d'aborder la recherche du nombre de classes k en amont d'un algorithme de classification par les k plus proches voisins.

4.4.1 Test d'unimodalité

Etre capable de comprendre comment sont regroupées les données dans un jeu de données multidimensionnel est une question fondamentale qui est souvent résolue avec les techniques de classification non supervisées encore appelées *clustering*. Cependant, sans avoir à pousser si loin le traitement, la contribution propose un test simple permettant de savoir rapidement si les données forment ou non un seul groupe cohérent. Dans le domaine des statistiques, on parle de distribution *unimodale*. Si dans la littérature, il existe déjà des tests d'unimodalité (le dip test [HH85] ou le Silverman test [Sil81]), ceux-ci sont cependant restreints aux données unidimensionnelles. Ce travail s'est intéressé aux données multidimensionnelles en proposant un test nommé FTU (pour *Folding Test of Unimodality*) qui présente comme caractéristiques de ne pas s'appuyer sur la distribution des données et de requérir un seul paramètre, un seuil p-value permettant d'affiner la confiance des résultats.

Le principe du test repose sur le pliage (*folding*) de la distribution. La figure 4.5(a) présente une distribution univariée bimodale pour laquelle la variance est vraisemblablement grande par la présence de ses deux modalités. Si l'on replie les deux modes l'un sur l'autre (cf. figure 4.5(b)), la variance résultante sera beaucoup plus faible que ce qui pourrait apparaître pour des données unimodales. L'approche consiste donc à : 1) trouver le pivot s^*, 2) opérer le pliage, 3) calculer la variance de la distribution une fois repliée, 4) la comparer avec la distribution initiale. Le *folding* ratio d'une distribution X est donc :

$$\varphi(X) = \frac{\mathrm{Var}|x - s^*|}{\mathrm{Var}X}$$



(a) Initial distribution (b) Folded distribution

FIGURE 4.5 – Principe du pliage sur une distribution univariée

Le calcul du pivot peut être estimé à l'aide d'une formule analytique qui peut se calculer, de manière incrémentale, à l'aide d'un algorithme de complexité linéaire

avec le nombre d'observations (pour plus de détails théoriques se référer à la publication [SFTL18]). Ce principe a pu être généralisé sur plusieurs dimensions $d \in \mathbb{N}$.

Le résultat intéressant est de pouvoir disposer d'un test d'aide à la décision permettant de savoir si une distribution X est unimodale ou non. La distribution uniforme étant considérée comme le pire cas d'unimodalité, il est possible de définir la *statistique de folding* $\Phi(X)$ de la manière suivante :

$$\Phi(X) = \frac{\varphi(X)}{\varphi(U_d)} = (1+d)^2 \,.\, \varphi(X)$$

avec U_d une variable aléatoire suivant la loi uniforme sur la boule unité en dimension d. Le test d'unimodalité FTU est donc défini de la manière suivante :

Si $\Phi(X) \geq 1$, la distribution X est unimodale
Si $\Phi(X) \leq 1$, la distribution X est multimodale

Plusieurs implémentations appelées `libfolding` du test FTU ont été réalisées en C++ [2], Python [3] ainsi qu'un package R appelé `Rfolding` [4] accepté par le CRAN.

Des expérimentations ont été menées dans le but de décider si un traitement de classification non supervisé est justifié ou non. Une première étude s'est intéressée aux données `Pokémon` disponibles sur le site `Kaggle` avec l'analyse des six compétences d'attaques. Sur un espace à six dimensions, il est difficile de savoir a priori si des classes se dégagent du jeu de données. Une comparaison a été menée avec des algorithmes reconnus de classification non supervisée capables de trouver automatiquement le nombre de clusters (cf. [SFTL18] pour les références des méthodes et le détail de l'expérimentation). Le test d'unimodalité FTU appliqué sur ces données retourne la valeur suivante : $\Phi(X) = 5.04$ indiquant clairement que la distribution est unimodale. Les résultats proposés par les autres méthodes ne donnent pas de consensus clair, le nombre de classes allant de 2 à plus de 5, selon les algorithmes, avec des classes différentes et présentant une valeur pour le coefficient de silhouette moyenne basse, voire proche de zéro, significative de classes peu distinctes. Ces résultats en faveur de l'unimodalité, confirment la réponse du test FTU et de son intérêt avant d'engager tout traitement de classification non supervisée.

2. Librairie `libfolding` en C++ : https ://github.com/asiffer/libfolding
3. Librairie `libfolding` en Python : https :// github.com/asiffer/python3-libfolding
4. Package `Rfolding` : https ://cran.r-project.org/package=Rfolding

4.4.2 Recherche du k dans la méthode des k plus proches voisins

Il est possible de s'appuyer sur les résultats du travail précédent pour trouver le nombre de classes de l'algorithme de classification non supervisée des k plus proches voisins (appelée k-means). L'idée consiste à développer ce que l'on appelle un *wrapper* autour de l'algorithme de classification. L'idée d'un *wrapper* autour de k-means n'est pas nouvelle et de nombreux wrappers existent déjà, on peut citer $X-$means [DPM00], $G-$means [HE03] ou *dip*-means [KL12] pour les données mutivariées, avec pour principe un processus itératif démarrant par une seule classe, k=1, puis augmentant successivement le nombre de classes jusqu'à la satisfaction d'un critère. Le principal inconvénient de ces méthodes est leur complexité qui limite leur passage à l'échelle. La contribution proposée dans la thèse d'Alban Siffer a pour originalité de s'intéresser à des données multivariées en gardant la philosophie de k-means, un algorithme simple de complexité raisonnable. L'algorithme de classification proposé, appelé Φ-means, s'appuie sur le test FTU. Le point essentiel garantissant son efficacité est son initialisation. Dans les méthodes classiques, les centres de classes sont choisis uniformément au hasard parmi les données ou à l'aide de probabilités. Dans notre approche, le pivot s^*, utilisé pour minimiser la variance de FTU, est réemployé par Φ-means pour le calcul des centres. Nous avons démontré que la concentration des données se situait à une certaine distance du pivot s^* au sein de laquelle il y avait la plus grande chance de trouver le centre. Les points à la distance $\rho = \mathbb{E}||X - s^*||$ sont donc de bons candidats pour initialiser k-means. Soit $X = \{X_1 \cdots X_n\}$ les données à séparer en deux classes suite au résultat du test FTU, on connait les distances $||X - s^*||$ (calculées pour le test). Le centre c_0 à distance ρ de s^* est le premier a être positionné, le second c_1 est symétrique par rapport au pivot. Les classes sont scindées jusqu'à ce qu'elles soient toutes unimodales.

L'algorithme $\Phi-$means a été implémenté dans une librairie Python et expérimenté sur des jeux de données en comparaison avec d'autres méthodes. Sur des données gaussiennes, $\Phi-means$ concurrence les méthodes réputées telles que $G-$means [HE03] ou $X-$means [DPM00]. Pour des données non-gaussiennes, $\Phi-$means surpasse la seule méthode pouvant traiter des données multivariées, *dip*-means [KL12], avec une vitesse 500 fois plus rapide pour la classification de 10000 observations. Ces résultats s'expliquent par la complexité linéaire de $\Phi - means$ alors que *dip*-means est de complexité quadratique.

Chapitre 5

Bilan et Perspectives

Sommaire

5.1 Bilan 107
5.2 Perspectives 111
5.2.1 Apprentissage de modèles de comportement interprétables 111
5.2.2 Modélisation des facteurs socio-économiques et de la résilience 113

Je conclurai ce mémoire en synthétisant les aspects qui m'ont paru essentiels dans les travaux menés avant de présenter quelques pistes de recherche que j'aimerais développer pour les années à venir.

5.1 Bilan

Les thèmes présentés dans ce mémoire m'ont conduit à m'intéresser aux systèmes dynamiques pour l'aide à la décision en gardant à l'esprit un point de vue orienté IA pour la représentation des connaissances et le raisonnement. Les problèmes d'aide à la décision s'adressent à des utilisateurs, qu'ils soient superviseurs de systèmes industriels pour le domaine du diagnostic, gestionnaires ou exploitants agricoles pour le management des agro-écosystèmes. Il s'agit de leur donner les clés, les plus compréhensibles possibles par l'explication des systèmes observés au regard des événements, pour les aider à prendre la meilleure décision.

La *modélisation qualitative* ou raisonnement qualitatif est une constante des approches présentées dans ce mémoire (en dehors du dernier chapitre sur la fouille de données). Le raisonnement qualitatif permet d'intégrer des masses de quantités hétérogènes, issues de plusieurs domaines et d'élargir ainsi le spectre des

modèles classiques par une abstraction des informations pertinentes. Si c'est le formalisme de référence dans le domaine du diagnostic à base de modèles, il a été plus difficile à se faire accepter dans le domaine de l'agro-écologie, coutumier des modèles mécanistes. Son principal intérêt, la manipulation des ensembles de valeurs qualitatives en tant qu'abstraction des valeurs numériques, réside dans l'*explicabilité* des résultats en offrant une représentation plus proche des modes de pensées des décideurs. Dans de nombreuses approches les systèmes qualitatifs peuvent être modélisés par des *systèmes à événements discrets*, c'est le choix opté pour nos travaux. Le comportement du système est déterminé par l'occurrence d'événements et sa représentation est donnée par des *automates*. Les événements portés par les transitions peuvent être de différente nature, des événements externes au système décrivant des actions désirables (des opérations d'utilisation ou de contrôle) ou indésirables qui sont en général non contrôlables. Ces événements peuvent également être d'origine interne et constituent alors la dynamique de chaque système. Les approches proposées reposent sur la modélisation d'un système par *composants en interaction*, qui présente l'avantage d'une facilité de construction du modèle surtout si l'on s'intéresse à des systèmes complexes. Chaque sous-partie du système peut être décrite individuellement et les interactions avec les voisins représentées par le biais d'événements synchronisés. Nous avons vu que la construction d'un modèle global reste compliquée pour établir le diagnostic et qu'une *approche décentralisée* reposant sur des diagnostic locaux était plus efficace. Nous avons pu cependant envisager un changement de topologie au fil du temps avec la définition des *systèmes reconfigurables*. Pour les autres approches, nous nous sommes appuyés sur un modèle global mais en utilisant une *représentation symbolique* qui décrit de manière plus compacte des ensembles d'états et réduisent ainsi considérablement la complexité des modèles.

Le *temps* est un aspect important des systèmes étudiés et représenté par un *ensemble discret* (et non dense comme dans les équations différentielles par exemple). Nous nous sommes appuyés sur ce temps, pour réduire la complexité par le calcul des diagnostic locaux pour des *fenêtres temporelles* sûres dans le cadre du calcul incrémental du diagnostic. Pour les autres travaux, le temps fait partie intégrante de la modélisation et est représenté à l'aide des *automates temporisés* sur lesquels on peut appliquer le *model-checking*. L'intérêt de ce formalisme est de pouvoir manipuler des états comportant des incertitudes temporelles dans l'esprit des intervalles d'Allen [All83] où des actions sont possibles sur des intervalles de temps, pouvant éventuellement se chevaucher. Cette représentation est particulièrement adaptée pour des systèmes que l'on connait mal et qui ne comportent pas d'effet de seuil. Le temps n'est cependant pas discrétisé dans notre approche, il reste une valeur continue croissante avec possible réinitialisation, laissant plus de finesse dans la description de la dynamique. Chaque composant du système peut disposer

de sa propre horloge temporelle et ainsi définir son comportement. La dynamique du modèle dépend donc à la fois des interactions entre les composants et des contraintes de temps. Les transitions sont instantanées mais peuvent intervenir dans un intervalle de temps contraint. Il en résulte un *non déterminisme* sur les évolutions possibles du système qui augmente la complexité d'exécution du modèle mais que nous n'avons pas cherché à supprimer, pour les problèmes de prédiction, puisqu'elle nous semblait intéressante pour aider la décision.

Des approches permettent de rajouter des informations quantitatives sur les automates, les probabilités étant les plus répandues, afin de supprimer ce non déterminisme. Cependant au regard de la complexité engendrée, en particulier lorsque plusieurs horloges cohabitent, et de la difficulté à inférer les probabilités du modèle initial, nous n'avons pas cherché à explorer cette voie. Pour les problèmes de recherche de stratégies, nous avons opté pour la valuation des actions à l'aide de *coûts* sur les états et transitions. Les coûts, qu'ils soient économiques, liés aux facilité de mise en œuvre, ou au risque, peuvent être aisément acquis auprès des experts. Nos travaux reposent essentiellement sur l'existence d'un modèle établi à partir de l'expertise ou abstraits à partir des modèles mécanistes, comme c'est le cas pour l'approche EcoMata. Cependant avec la disponibilité croissante des données et l'augmentation de la vitesse de calcul, nous aborderons progressivement l'apprentissage automatique de modèles.

Les problèmes d'*aide à la décision* que nous avons traité sont les suivants : la recherche de panne pour le diagnostic, la supervision qui fournit les trajectoires cohérentes avec un ensemble d'observations, la prévision pour des scénarios prédictifs *what if ?*, la recommandation d'actions avec la recherche de stratégies *what for ?*. Dans les domaines du *diagnostic* ou de la *supervision*, l'acteur n'est pas impliqué dans le processus en dehors de la création du modèle. Le domaine concerne principalement les systèmes industriels et les informations en entrée des algorithmes sont issues des capteurs. Le résultat peut être selon, l'identification de la panne et le composant concerné, la suite d'événements expliquant le comportement anormal. Pour les problèmes de management, en particulier pour la *prédiction*, l'acteur est beaucoup plus actif à toutes les étapes du processus. Le modèle n'est pas un et unique mais construit à l'aide des hypothèses d'usage. L'utilisateur a le choix d'interroger le modèle à l'aide d'un *langage de haut-niveau* en construisant le raisonnement qui lui permet d'établir son opinion. Pour les approches de fouille données, les travaux se limitent aujourd'hui à la prédiction, même si l'on espère affiner la prédiction en tenant compte du comportement réel (ici les truies) dans un esprit de supervision. L'aide à la *recommandation d'actions* a été traitées selon deux types de problèmes : l'atteignabilité et la sûreté. Le résultat fourni est beaucoup plus guidé en proposant une suite d'actions datées permettant

d'atteindre l'objectif visé. Ces deux approches me paraissent complémentaires, la prédiction est un outil essentiel pour comprendre le fonctionnement du système, ou pour aider à la concertation, mais peu suffisant pour établir un plan d'actions, le décideur ne peut itérer indéfiniment jusqu'à trouver la bonne solution. C'est alors qu'intervient la recherche de stratégies plus efficace et plus pragmatique.

Les travaux effectués ont porté sur la définition d'algorithmes maitrisant la complexité pour autoriser leur utilisation sur des applications réelles. Nous sommes partis des systèmes dynamiques événementiels sur lesquels nous avons progressivement ajouté le temps puis le coût. Ces nouvelles dimensions accroissent la complexité mais permettent une meilleure connaissance des systèmes étudiés. Nos algorithmes ont utilisé des techniques de *réduction de cette complexité* comme les chaines d'automates, le calcul incrémental mais aussi les techniques de model-checking, un solveur logique tel qu'ASP, ou la synthèse de contrôleur reposant sur une représentation symbolique du système. Pour les travaux en apprentissage automatique, il s'agit de maitriser la complexité des données sans hypothèse sur leur distribution comme pour la détection des anomalies ou dans leur dimensionnalité pour l'unimodalité.

Enfin, les *applications* ont été le moteur de ces travaux en particulier pour la partie dédiée à la gestion des écosystèmes. On peut citer les différents écosystèmes marins, de la pêche de subsistance à Ouvéa, à la pêche à la coquilles en rade de Brest, en passant par la mer du Nord, la gestion des pâturages ou l'alimentation des truies allaitantes.

Mais aussi des applications dans lesquelles je me suis investie et que je n'ai pas citées dans ce mémoire, parce qu'en marge des thèmes présentés ici, la modélisation d'un récif corallien touché par un cyclone à partir d'images de télédétection [SAL07], la dynamique des risques de pollution pour aider à la gestion de l'eau douce [MLTL05], la gestion intégrée des zones côtières en Nouvelle-Calédonie à l'aide des cartes cognitives décrivant la vision des acteurs [LFFP09]. Lorsqu'il s'agit de proposer une nouvelle approche à partir d'une application réelle, le chemin est parfois long et tortueux et ne mène pas toujours à un résultat satisfaisant. C'est le cas pour la modélisation spatio-temporelle des carabes-adventices, lors d'une collaboration pourtant très dynamique avec l'INRA, pour laquelle nous n'avons pas su trouver la bonne abstraction pour permettre un passage à l'échelle sur un territoire intéressant vis-à-vis du problème. Ces cas, pourtant rares, illustrent la complexité de modéliser les agro-écosystèmes mais c'est également ce qui les rend si motivants, en dehors des enjeux écologiques, pour imaginer des solutions nouvelles.

5.2 Perspectives

De nombreuses perspectives de recherches sont ouvertes à la suite de ces travaux. Elles s'intègrent aux thématiques de l'équipe INRIA Lacodam pour laquelle nous redéfinissons actuellement les objectifs pour une nouvelle période de quatre ans.

5.2.1 Apprentissage de modèles de comportement interprétables

Jusqu'à présent, la modélisation des comportements du système pouvait être considérée comme un "art" qui s'appuyait sur des spécialistes du domaine qui devaient être disponibles et désireux de s'investir. Avec la disponibilité des grands volumes de données, on peut imaginer la construction automatique de ces modèles dans un objectif très large de l'aide à la décision en ciblant toujours les acteurs et non les machines. Mon objectif consiste à apprendre des *modèles de comportement* de systèmes complexes pour les utiliser dans un processus d'exploration qu'il soit en ligne ou non.

Modèles interprétables
Malgré leur efficacité, le principal revers des techniques d'apprentissage automatique est leur aspect "boite noire". Dans les domaines où l'homme doit prendre des décisions fondées sur de grandes quantités de données, l'*interprétabilité* est fondamentale et fait aujourd'hui partie des attentes des utilisateurs (et du RGPD). L'interprétabilité d'un modèle repose sur la notion d'explication qui peut être vue comme une interface compréhensible par l'homme [GMT+19]. Des mesures permettent aujourd'hui d'évaluer cette interprétabilité comme la transparence, la précision, la fidélité, la stabilité, etc. L'interprétabilité pouvant même être perçue comme une nouvelle "science" [GBY+19]. Il est possible d'envisager l'interprétabilité de manière globale, dans le sens où la logique complète du modèle peut être comprise, ou locale, pour répondre à un problème précis de prédiction, avec l'homme qui intervient directement dans la boucle d'optimisation de l'interprétabilité [LRG+18]. Si les modèles doivent être interprétables, il en est de même des interactions de l'utilisateur vis-à-vis des ces modèles, que ce soit les requêtes ou les sorties du modèle. Peu de travaux se consacrent à l'interprétabilité de modèles complexes, et même les classifieurs à base de règles, reconnus comme les plus interprétables, se heurtent à cette complexité lorsque la taille des règles tend à croitre. Le compromis entre efficacité et interprétatibilité devient le nouvel enjeu de l'apprentissage automatique [Rud18]. C'est par cette voie, qu'il me semble le plus naturel et intéressant de faire le lien entre mes précédents travaux, où la modélisation qualitative visait les mêmes objectifs, et l'apprentissage automatique de modèles. Deux types d'approches sont généralement proposés pour tenter de

répondre à ce problème. La première consiste à rendre interprétables les "black-box", par une méthode appelée *reverse engineering*, comme apprendre des règles, à partir d'un réseau de neurones par exemple[Hai16]. La seconde consiste à apprendre directement un modèle interprétable depuis les données.

Modèles temporels

Le but serait d'apprendre des modèles temporels qui expriment le comportement ou la dynamique des systèmes. L'intérêt des modèles de comportement est qu'ils peuvent ensuite être utilisés pour des tâches telles que la supervision des systèmes à partir d'un flot d'observations pour déterminer, par exemple, si le comportement est conforme ou non. Ils permettent par ailleurs de répondre plus aisément aux questions d'interprétabilité par une représentation, même abstraite, du système analysé. La première étape s'intéresse à l'extraction de propriétés de haut niveau permettant de représenter le changement d'état du système. Les modèles les plus connus sont les chroniques, suite aux premiers travaux de Dousson [DD99], qui sont efficaces pour le diagnostic [STMLC14] et ont été appliquées à de nombreux domaines. Une perspective à court ou moyen terme serait de m'intéresser aux formalismes de systèmes à événements discrets temporisés comme les automates ou les réseaux de Pétri qui peuvent être utilisés pour la vérification et la validation à l'aide de méthodes formelles. Le premier défi à relever consiste à trouver la bonne abstraction du modèle puis à apprendre les comportements intéressants, certaines techniques de généralisation pouvant être employées afin de réduire la taille du modèle appris [LLMP08]. L'inférence des contraintes temporelles des comportements est un point clé et résolu par deux catégories d'apprentissage : l'apprentissage actif qui interroge le système, pour recevoir des informations complémentaires ou l'utilisateur, pour valider les hypothèse, et l'apprentissage passif qui doit se satisfaire des données disponibles. Des algorithmes existent pour l'apprentissage d'automates temporisés [LZVW19, PMM17] mais reposent sur des hypothèses fortes, en particulier que les états peuvent être directement reconnus à partir données issues des capteurs. Ces algorithmes peuvent alors se comparer, dans la génération des automates, à que nous avions pu proposer dans EcoMata avec des états issus d'un modèle mécaniste. En réalité, sur des données complexes, les états et transitions temporelles associées sont déjà bien plus complexes à identifier. Une première approche pourrait s'appuyer sur des techniques de détection des patterns éventuellement temporels qui est un thème porteur de l'équipe Lacodam, pour inférer ces états et générer les systèmes à événements discrets. Une perspective à plus long terme serait de pouvoir inférer ces comportements en ligne en bénéficiant de leur efficacité. Ces modèles de comportement peuvent s'avérer utiles en agro-écologie pour représenter les itinéraires techniques par exemple, à partir des historiques de traitements des parcelles, ou la supervision des animaux pour

assurer leur bien-être en identifiant les attitudes anormales (agressivité, isolement).

Séries temporelles
Pour apprendre ces modèles de comportement, les séries temporelles sont les sources de données les plus intéressantes. L'analyse de séries temporelles est un domaine en pleine croissance, le principal thème relevant surtout de la classification supervisée ou non. Beaucoup d'approches utilisent des architectures à base de réseaux neuronaux convolutifs tels que réseaux de neurones récurrents. L'inconvénient majeur est la nécessité de disposer d'un volume très important de données d'entrainement et l'annotation des données doit être effectuée par des experts du domaine parfois difficiles à trouver. Apprendre les modèles de comportement à partir de séries temporelles constitue une perspective intéressante qui va dans le sens des travaux sur les pattern temporels en particulier l'approche des patterns prédictifs de Batal qui s'appuie sur une représentation abstraite des patterns temporels appris [BFH+12].

5.2.2 Modélisation des facteurs socio-économiques et de la résilience

Le couplage des modèles de gestion des écosystèmes et des modèles socio-économiques offre une vision élargie qui évite l'isolement de l'écosystème de la réalité socio-économique. Certaines intégrations assemblent aujourd'hui des modèles indépendants, s'exécutant en parallèle, qui peuvent s'avérer suffisant lorsque les modèles économiques et environnementaux sont linéaires. Une modélisation unifiée est au contraire appropriée lorsque les impacts environnementaux ont des effets économiques ou lorsque les décisions économiques ou sociales intègrent des politiques environnementales. Une première perspective à court terme est l'intégration des facteurs socio-économiques dans la modélisation de l'écosystème sous une forme plus homogène que la représentation sous forme de règles actuellement proposée. Cette représentation vise à décrire les variations du marché et les habitudes de consommation. Si les méthodes reposant des approches bayésiennes peuvent satisfaire les critères demandés, elles nécessitent cependant de fortes hypothèses sur la distribution des données en entrée du modèle [BEK+15]. Une perspective à court terme serait d'introduire des modèles à coûts pour représenter ces facteurs socio-économiques. Des travaux récents montrent que pour les réseaux de Pétri temporels à coûts, le problème de l'atteignabilité pour un coût optimal peut être résolu pour une sous-classe de modèles [BLRS18], ce qui nous permettrait également d'évaluer les potentialités de ce formalisme.

En dernier point, j'aimerais m'intéresser au problème de la résilience des écosystèmes. La résilience intègre la capacité du système à supporter les pressions tout en préservant ses fonctions et son aptitude à retrouver son état de stabilité. Cette notion de résilience s'explique souvent à l'aide d'un effet de seuil, que ne peut franchir un écosystème s'il veut retrouver ses fonctionnalités. Les stratégies de management des écosystèmes doivent donc anticiper et s'adapter afin d'éviter de franchir ce seuil et faire basculer le système dans une situation d'épuisement. Malheureusement, la définition des seuils est difficile à identifier s'ils n'ont pas été franchis, aussi les écologues par leur expérience peuvent s'appuyer sur les modèles qualitatifs, moins contraints, pour proposer des solutions. Prendre en considération la résilience des écosystèmes consiste à trouver la stratégie la plus économiquement viable tout en garantissant la protection de l'écosystème. La résilience des écosystèmes a déjà fait l'objet de nombreux paradigmes de modélisation depuis les années 70 [MB14] mais l'urgence environnementale nous amène à reconsidérer autrement nos liens avec la nature. On constate que les travaux actuellement menés sont très spécialisés et cantonnés à chaque sphère disciplinaire [FGA15]. Les modélisations proposées par nos travaux sur les écosystèmes marins pourraient servir de base de travail pour proposer un modèle plus riche permettant d'intégrer à la fois les aspects théoriques de la résilience et l'intégration des données de capteurs [HHH$^+$15, HR06].

Bibliographie

[ABA17] J. Amara, B. Bouaziz, and A. Algergawy. A deep learning-based approach for banana leaf diseases classification. *Lecture Notes in Informatics (BTW)*, pages 79–88, 2017.

[ABP06] F. Angiulli, S. Basta, and C. Pizzuti. Distance-based detection and prediction of outliers. *IEEE transactions on knowledge and data engineering*, 18(2) :145–160, 2006.

[AC09] L. Amgoud and C. Cayrol. Using arguments for making and explaining decisions. *International Journal of Automated Reasoning*, 29(2) :125–169, 2009.

[ACD93] R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1) :2–34, 1993.

[Ack79] R. L. Ackoff. The future of operational research is past. *Journal of the Operational Research Society*, 30(2) :93–104, 1979.

[AD90] R. Alur and D. Dill. Automata for modeling real-time systems. In *17th International Colloquium on Automata, Languages and Programming (ICALP90)*, pages 322–335, 1990.

[AD94] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126 :183–235, 1994.

[All83] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications ACM*, 26(11) :832–843, 1983.

[Alt80] S.L. Alter. *Decision Support Systems : Current Practice and Continuing Challenge*. Addison-Wesley, 1980.

[ALTP01] R. Alur, S. La Torre, and G.J. Pappas. Optimal paths in weighted timed automata. In *HSCC-2001*, pages 49–62. Springer, 2001.

[Amg00] L. Amgoud. Modelling dialogues using argumentation. In *International Conference on Multi-Agent Systems (ICMASS'00)*, pages 31–38, 2000.

[AMP95] E. Asarin, O. Maler, and A. Pnueli. Symbolic controller synthesis for discrete and timed systems. In *Hybrid Systems II, LNCS 999*, pages 1–20. Springer, 1995.

[AMPS98] E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *IFAC Symposium on System Structure and Control*, pages 469–474. Elsevier Science, 1998.

[AP09] L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173 :413–436, 2009.

[APKH16] A. Aliev, W Pedrycz, V. Kreinovich, and O. Huseynov. The general theory of decisions. *Information Sciences*, 327 :324–331, 2016.

[BAEF15] N. Ben Amor, F. Essghaier, and H. Fargier. Décision collective sous incertitude possibiliste. Principes et axiomatisation. *Revue d'Intelligence Artificielle*, 29(5) :515–542, 2015.

[BBD+14] C. Briguez, M. Budan, C. Deagustini, A. Maguitman, M. Capobianco, and G. Simari. Argument-based mixed recommenders and their application to movie suggestion. *Expert Systems with Applications*, 41 :6467–6482, 2014.

[BBF+01] B. Berard, M. Bidoit, A. Finkel, F. Laroussine, A Petit, P. Schnoebelen, and P. McKenzie. *Systems and Software Verification. Model-Checking Techniques and Tools*. Springer-Verlag, 2001.

[BBLPB16] N. Becu, P. Bommel, C. Le Page, and F. Bousquet. Cormas, une plate-forme multi-agent pour concevoir collectivement des modèles et interagir avec les simulations. In *Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*, 2016.

[BCD+07] G. Behrmann, A. Cougnard, A. David, E. Fleury, K.G. Larsen, and D. Lime. Uppaal-tiga : Time for playing games ! In *CAV-2007*, pages 121–125, 2007.

[BCFL04] P. Bouyer, F. Cassez, E. Fleury, and K.G. Larsen. Optimal strategies in priced timed game automata. In *FSTTCS-2004*, pages 148–160. LNCS 3328, 2004.

[BDH74] A. Balkema and L. De Haan. Residual life time at great age. *The Annals of probability*, pages 792–804, 1974.

[BDPP09] D. Bouyssou, D. Dubois, H. Prade, and M. Pilot. *Decision-Making Process, Concepts and Methods*. Wiley, 2009.

[BEK+15] L. Blume, D. Easley, J. Kleinberg, R. Kleinberg, and E. Tardos. Introduction to computer science and economic theory. *Journal of Economic Theory*, 156 :1–13, 2015. Computer Science and Economic Theory.

[BFH+12] I. Batal, D. Fradkin, J. Harrison, F. Moerchen, and M. Hauskrecht. Mining recent temporal patterns for event detection in multivariate time series data. In *ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining (KDD'12), pages 280–288, 2012.

[BGK04] Y.-M Bozec, D. Gascuel, and M. Kulbicki. Trophic model of lagoonal communities in a large open atoll (Uvea, Loyalty Islands, New Caledonia). *Aquatic Living Resources*, 17 :151–162, 2004.

[BGST06] J. Beirlant, Y. Goegebeur, J. Segers, and J. Teugels. *Statistics of extremes : theory and applications*. John Wiley & Sons, 2006.

[BH01] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128 :203–235, 2001.

[BHD+06] L. Börjeson, M. Höjer, K.-H. Dreborg, T. Ekvall, and G. Finnveden. Scenario types and techniques : Towards a user's guide. *Futures*, 38 :723–739, 2006.

[BHO14] S. Bensalem, K. Havelund, , and A. Orlandini. Verification and validation meet planning and scheduling. *International Journal on Software Tools for Technology Transfer*, 16(1) :1–12, 2014.

[BLR04] G. Behrmann, K.G. Larsen, and J.I. Rasmussen. Priced timed automata : Algorithms and applications. In *FMCO-2004*, pages 162–182. Springer-Verlag, 2004.

[BLRS18] H. Boucheneb, D. Lime, O. Roux, and C. Seidner. Optimal-cost reachability analysis based on time petri nets. In *18th International Conference on Application of Concurrency to System Design, (ACSD)*, pages 30–39, 2018.

[BM15] B. Boussif and Ghazel M. Diagnosability analysis of input/output discrete-event systems using model-checking. In *5th IFAC Workshop on Dependable Control of Discrete Systems*, 2015.

[BMPV06] D. Bouyssou, T. Marchant, M. Pirlot, and P. Vincke. *Evaluation and Decision Models with Multiple Criteria*. Springer, 2006.

[BMS00] C. Barral, S. McIlraith, and T. C. Son. Formulating diagnostic problem solving using an action language with narratives and sensing. In *International Conference on Principle of Knowledge Representation and Reasoning (KR2000)*, pages 311–322, 2000.

[Bou94] G. Boutilier. Toward a logic for qualitative decision theory. In *4th International Conference on Knowledge Representation and Reasoning (KR'94)*, pages 75–86, 1994.

[Bou03] D. Bouyssou. La crise de la recherche opérationnelle : 25 ans après. *Mathématiques et Sciences Humaines*, 2003.

[Bow03] K. Bowen. Sixty years of Operational Research. *European Journal of Operational Research*, 153 :618–623, 2003.

[BP04] F. Bousquet and C. Le Page. Multi-agent simulations and ecosystem management : a review. *Ecological Modelling*, 176(3) :313 – 332, 2004.

[BPDT17] F. Basile, Chiacchio P., and G. De Tommasi. Diagnosability analysis of patterns on bounded labeled prioritized petri nets. *Discrete Event Dynamic Systems*, 27(1) :143–180, 2017.

[BT96] R. I. Brafman and M. Tennenholtz. On the foundations of qualitative decision theory. In *AAAI 96, Portland, Oregon, August 4-8, 1996, Volume 2.*, pages 1291–1296, 1996.

[BTMJ13] J.-R. Bourguet, R. Thomopoulos, M.-L. Mugnier, and Abécassis J. An artificial intelligence-based approach to deal with argumentation applied to food quality in public heath policy. *Artificial Intelligence Review*, 40 :4539–4546, 2013.

[Bul14] N. Bulling. A Survey of Multi-Agent Decision Making. *Künstliche Intelligenz*, 28(3) :147–158, 2014.

[CBK09] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection : A survey. *ACM computing surveys (CSUR)*, 41(3) :1–72, 2009.

[CFF+10] A. Cesta, A. Finzi, S. Fratini, A. Orlandini, and E. Tronci. Analyzing flexible timeline-based plans. In *ECAI 2010*, pages 471–476, 2010.

[CGP02] E.M. Clarke, O. Grumberg, and A. Peled, D. *Model-Checking*. MIT Press, 2002.

[CKMS19] P. Cabalar, R. Kaminski, P. Morkisch, and T. Schaub. telingo = ASP + time. In *Logic Programming and Nonmonotonic Reasoning - 15th International Conference, LPNMR 2019*, pages 256–269, 2019.

[CL01] M.-O. Cordier and C Largouët. Using model-checking techniques for diagnosing discrete-event systems. In *12th International Workshop on Principles of Diagnosis DX'2001*, pages 39–46, 2001.

[CL08] C. Cassandras and S. Lafortune. *Introduction to discrete event systems, second edition*. Kluwer Academic Publishers, 2008.

[ČLMM14] P. Čermák, A. Lomuscio, F. Mogavero, and A. Murano. Mcmas-slk : A model checker for the verification of strategy logic specifications. In *Computer Aided Verification*, pages 525–532. Springer International Publishing, 2014.

[CP87] P.T. Cox and T. Pietrzykowski. General diagnosis by abductive inference. In *IEEE Symposium on Logic Programming)*, pages 183–189, 1987.

[CPRT03] A. Cimatti, M. Pistore, M. Roveri, and P. Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1-2) :35–84, 2003.

[CTC+12] R. Craven, F. Toni, C. Cadar, A. Hadad, and M. Williams. Efficient argumentation for medical decision-making. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning (KR'12)*, pages 598–602, 2012.

[Dan90] G. B. Dantzig. *A History of Scientific Computing.* ACM, New York, NY, USA, 1990.

[Dav82] R. Davis. Expert systems : Where are we ? and where do we go from here ? *AI Magazine*, 3(2) :3–22, 1982.

[DD99] C. Dousson and T. V. Duong. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, (IJCAI)*, pages 620–626, 1999.

[DD11] Albert David and Sébastien Damart. Bernard Roy et l'aide multicritère à la décision. *Revue française de gestion*, 37(214) :15–28, 2011.

[DFP03] D. Dubois, H. Fargier, and P. Perny. Qualitative decision theory with preference relations and comparative uncertainty : An axiomatic approach. *Artificial Intelligence*, 48 :219–260, 2003.

[DG94] A. Darwiche and M. Goldszmidt. On the relation between kappa calculus and probabilistic reasoning. In *10th Conference on Uncertainty in AI*, pages 1145–153, 1994.

[DHR03] J. Dambacher, W. Hiram, and P. Rossignol. Qualitative predictions in model ecosystems. *Ecological Modelling*, 161 :79–93, 2003.

[DLR05] J. Dambacher, R. Levins, and P. Rossigol. Life expectancy change in perturbed communities : Derivation and qualitative analysis. *Mathematical Biosciences*, 197 :1–12, 2005.

[DLT00] R. Debouk, S. Lafortune, and D. Tenekezis. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete event dynamic systems*, 10 :33–86, 2000.

[DP95] D. Dubois and H. Prade. Possibility theory as a basis for qualitative decision theory. In *IJCAI'95*, pages 1924–1932, 1995.

[DPM00] D. Dan Pelleg and A. W. Moore. X-means : Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 727–734, 2000.

[DR05] S. Damart and S. Roy. Débat public et expertise : entre légitimité et rationalité. *Gérer et Comprendre*, 82 :91–110, 2005.

[DT99] J. Doyle and R.H. Thomason. Background to qualitative decision theory. *AI Magazine*, 20 :55–68, 1999.

[ELN+08] M. Elahi, K. Li, W. Nisar, X. Lv, and H. Wang. Efficient clustering-based outlier detection algorithm for dynamic data stream. In *Fifth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, volume 5, pages 298–304. IEEE, 2008.

[FGA15] L. Fraccascia, I. Giannoccaro, and V. Albino. Resilience of complex systems : State of the art and directions for future research. *Complexity*, 2018(9), 2015.

[Fre84] R.E. Freeman. *Strategic Management : A stakeholder Approach*. Boston, MA, 1984.

[FT28] R. Fisher and L. Tippett. Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 24, pages 180–190, 1928.

[FTG+17] E. Ferretti, L. Tamargo, A. Garcia, M. Errecalde, and G. Simari. An approach to decision making based on dynamic argumentation systems. *Artificial Intelligence*, 242 :107–131, 2017.

[GA08] A. Grastien and A. Anbulagan. Incremental diagnosis of DES by satisfiability. In *European Conference on Artificial Intelligence (ECAI-08)*, pages 787–788, 2008.

[GA13] A. Grastien and A. Anbulagan. Diagnosis of discrete event systems using satisfiability algorithms : a theoretical and empirical study. *IEEE Transactions on Automatic Control (TAC)*, 58(12) :3070–3083, 2013.

[Gai15] P. Gaignon. Modélisation et exploration d'un écosystème à dynamique spatiale par des outils de model-checking. Master's thesis, Agrocampus Ouest, 2015. Rapport de stage de Master 2, spécialité statistique.

[GBY+19] L. Gilpin, D. Bau, B. Yuan, A. Bajawal, M. Specter, and L. Kagal. Explaining explanations : An approach to evaluating interpretability of machine learning. In *IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, 2019.

[GC06] A. Grastien and M.-O. Cordier. Exploiting independence in a decentralized and incremental approach of diagnosis. In *17th International Workshop on Principles of Diagnosis (DX'06)*, pages 61–69, 2006.

[GCL04] A. Grastien, M.-O. Cordier, and C Largouët. Extending decentralized discrete-event modelling to diagnose reconfigurable systems. In *15th*

International Workshop on Principles of Diagnosis (DX'04), pages 75–80, 2004.

[GCL05a] A. Grastien, M.-O. Cordier, and C Largouët. Automata slicing for diagnosis discrete-event systems with partially ordered observations. In *Congress of the Italian Association for Artificial Intelligence (AIIA'05)*, pages 270–281, 2005.

[GCL05b] A. Grastien, M.-O. Cordier, and C Largouët. First steps towards incremental diagnosis of discrete-event systems. In *Canadian AI'05, Victoria, Canada*, pages 170–181, 2005.

[GCL05c] A. Grastien, M.-O. Cordier, and C Largouët. Incremental diagnosis of discrete-event systems. In *16th International Workshop on Principles of Diagnosis (DX'05)*, pages 1564–1565, 2005.

[GCLP03] A. Grastien, M.-O. Cordier, C Largouët, and Y. Pencolé. Efficient trajectories computing using inversibilities properties. In *14th International Workshop on Principles of Diagnosis (DX'03)*, pages 93–98, 2003.

[GCRS13] A. Garcia, C. Chesñevar, N. Rotstein, and G. Simari. Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. *Expert Systems with Applications*, 40 :3233–3247, 2013.

[GD01] F. Guerrin and J. Dumas. Knowledge representation and qualitative simulation of salmon redd functioning. part i : qualitative modeling and simulation. *Biosystems*, 59(2) :75–84, 2001.

[GISM+18] K. Gibert, J. Izquierdo, M. Sànchez-Marrè, S. Hamilton, I. Rodríguez-Roda, and G. Holmes. Which method to use ? An assessment of data mining methods in Environmental Data Science. *Environmental Modelling and Software*, 110 :3–27, 2018.

[GKK+11] M. Gebser, R. Kaminski, Kaufmann, Ostrowski B., Schaub M., and M. T., Schneider. Potassco : The potsdam answer set solving collection. *Discrete Event Dynamic Systems*, 24(2) :107–124, 2011.

[GLG+19] R. Gauthier, C. Largouët, C. Gaillard, L. Cloutier, F. Guay, and J.-Y. Dourmad. Dynamic modeling of nutrient use and individual requirements of lactating sows. *Journal of Animal Science*, 97(7), 2019.

[GMT+19] R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti. A survey of methods for explaining black box models. *ACM Computing Survey (CSUR)*, 51, 2019.

[Gne43] B. Gnedenko. Sur la distribution limite du terme maximum d'une série aléatoire. *Annals of mathematics*, pages 423–453, 1943.

[GP05] P. Gmytrasiewicz and Doshi P. A framework for Sequential Planning in Multi-Agent Settings. *Journal of Artificial Intelligence Research*, 24 :49–79, 2005.

[GPS17] H.-E. Gougam, Y. Pencolé, and A. Subias. Diagnosability analysis of patterns on bounded labeled prioritized petri nets. *Discrete Event Dynamic Systems*, 27(1) :143–180, 2017.

[Gra05] A. Grastien. *Diagnostic décentralisé et en ligne de systèmes à événements discrets reconfigurables*. PhD thesis, Université de Rennes I, France, 2005.

[Gra09] A. Grastien. Symbolic testing of diagnosability. In *20th International Workshop on Principles of Diagnosis (DX-09)*, pages 131–138, 2009.

[GT99] F. Giunchiglia and P. Traverso. Planning as model checking. In *ECP'99*, pages 1–20, 1999.

[Gue91] F. Guerrin. Qualitative reasoning about an ecological process : interpretation in hydroecology. *Ecological Modelling*, 59(2) :165–201, 1991.

[Hai16] T. Hailesilassie. Rule extraction algorithm for deep neural networks : A review. *International Journal of Computer Science and Information Security*, 14(7) :376–381, 2016.

[HE03] G. Hamerly and G. Elkan. Learning the k in k-means. In *Advances in Neural Information Processing Systems, NIPS'03*, pages 281–288, 2003.

[HH85] J.A Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13(1) :70–84, 1985.

[HHH+15] M. Hipsey, D. Hamilton, P. Hanson, C. Carey, J. Coletti, J. Read, B. Ibelings, F. Valesini, and J. Brookes. Predicting the resilience and recovery of aquatic systems : A framework for model evolution within environmental observatories. *Water Resources Research*, 51(9) :7023–7043, 2015.

[HKP12] J. Han, M. Kamber, and J. Pei. *Data Mining, Concepts and Techniques*. Morgan Kauffmann, 3 edition, 2012.

[HNSY94] T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2) :193–244, 1994.

[HR06] C. Hawes and C. Reed. Theoretical steps towards modelling resilience in complex systems. In *Proceedings of the 6th International Conference on Computational Science and Its Applications - Volume Part I (ICCSA)*, pages 644–653, 2006.

[HSK+12] A. Hansen, A. Strathe, E. Kebreab, J. France, and P. Theil. Predicting milk yield and composition in lactating sows : A bayesian approach. *Journal of Animal Science*, 90(7) :2285–2298, 2012.

[HWP+18] D. Hooftman, S. Willcock, C. Prato, J. Bullock, K. Bagstad, S. Balbi, B. Voigt, I. Athanasiadis, F. Villa, S. Sciandrello, G. Signorello, A. Marzo, and J. Martínez-López. Machine learning for ecosystem services. *Ecosystem Services*, 33 :165–174, 2018.

[IAS+18] R. Ip, L.-M. Ang, K. Seng, J. Broster, and J. Pratley. Big data and machine learning for crop protection. *Computers and Electronics in Agriculture*, 151 :376–383, 2018.

[JPM+17] S. Janssen, C. Porter, A. Moore, I. Athanasiadis, I. Foster, J. Jones, and J. Antle. Towards a new generation of agricultural system data, models and knowledge products : Information and communication technology. *Agricultural Systems*, 155 :200–212, 2017.

[JW87] De Kleer J. and B. C. Williams. Diagnosis multiple faults. *Artificial Intelligence*, 32(1) :97–130, 1987.

[Kal10] B. Kalantari. Herbert A. Simon on making decisions : enduring insights and bounded rationality. *Journal of Management History*, 16(4) :509–520, 2010.

[Kee87] G.W. Keen. Decision support system : The next decade. *Decision Support Systems*, 3(3) :253–265, 1987.

[KKPB17] A. Kamilaris, A. Kartakoullis, and F. X. Prenafeta-Boldú. A review on the practice of big data analysis in agriculture. *Computers and Electronics in Agriculture*, 143 :23–37, 2017.

[KL12] A. Kalogeratos and A. Likas. Dip-means : an incremental clustering method for estimating the number of clusters. In *Advances in Neural Information Processing Systems, NIPS'12*, pages 2402–2410, 2012.

[KMH00] L. Khatib, N. Muscettola, and K. Havelund. Verification of plan models using UPPAAL. In *FAABS 2000*, pages 114–122, 2000.

[KPB18a] A. Kamilaris and F. Prenafeta-Boldú. Deep learning in agriculture : A survey. *Computers and Electronics in Agriculture*, 147 :70–90, 2018.

[KPB18b] A. Kamilaris and F. Prenafeta-Boldú. A review of the use of convolutional neural networks in agriculture. *Journal of Agricultural Science*, pages 1–11, 2018.

[KR13] E. Keogh and T. Rakthanmanon. Fast shapelets : A scalable algorithm for discovering time series shapelets. In *Proceedings of the*

13th SIAM International Conference on Data Mining, USA., pages 668–676, 2013.

[KR16] V. Kaleeswaran and R. Rathipriya. A study on data mining approches for agricultural intelligence. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(4) :453–456, 2016.

[KSM78] P. Keen and M. Scott-Morton. *Decision Support Systems : an organizational perspective*. Wesley Publishing, 1978.

[KSS+17] A. Kumar, B. Sah, A.R. Singh, Y. Deng, X. He, P. Kumar, and R.C. Bansal. A review of multi criteria decision making (MCDM) towards sustainable renewable energy development. *Renewable and Sustainable Energy Reviews*, 69(November 2016) :596–609, 2017.

[Kui94] B. Kuipers. *Qualitative reasoning : modeling and simulation with incomplete knowledge*. MIT Press Cambridge, MA, USA, 1994.

[Laf16] F. Laferrière. Diagnostic de systèmes à événements discrets par answer set programming. Master's thesis, Université d'Angers, 2016. Rapport de stage de Master 1 Informatique.

[LBM+18] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis. Machine learning in agriculture : A review. *Sensors (Switzerland)*, 18(8) :1–29, 2018.

[LBZ16] C. Largouët, Y.-M. Bozec, and Y. Zhao. Integrating socio-economic drivers in an explicit-time, qualitative fisheries model : EcoMata. In *Understanding Marine Socio-ecological Systems including the human dimension in integrated ecosystem assessment MSEAS 2016*, May 2016.

[LC00] C. Largouët and M.-O. Cordier. Timed automata model to improve the classification of a sequence of images. In *European Conference of Artificial Intelligence (ECAI'00)*, pages 156–160, 2000.

[LCB+11] C. Largouët, M.-O. Cordier, Y.-M. Bozec, Y. Zhao, and G. Fontenelle. Use of timed automata and model-checking to explore scenarios on ecosystem model. *Environmental Modelling and Software*, 30 :123–138, 2011.

[LF14] R. Laxhammar and G. Falkman. Online learning and sequential anomaly detection in trajectories. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 36(6) :1158–1173, 2014.

[LFFP09] C. Largouet, G. Fontenelle, J. Ferraris, and D. Poignonec. Les cartes cognitives : un outil pertinent pour initier un processus de gestion

intégrée des zones côtières pour des écosystèmes lagonaires sous pression : une application en nouvelle-calédonie. In *11th Pacific Science Intercongress*, 2009.

[LFL04] M. Léopold, J. Ferraris, and P. Labrosse. Assessment of the reliability of fish consumption as an indicator of reef fish catch in small pacific islands : the example of ouvéa island in new caledonia. *Aquatic Living Resources*, 17 :119–127, 2004.

[LJC89] P. Levine and Pomerol J.-C. *Systèmes interactifs d'aide à la décision et systèmes experts*. Hermès, 1989.

[LJC95] P. Levine and Pomerol J.-C. The role of the decision maker in dsss and representation levels. In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, pages 42–51, 1995.

[LKJ+16] R. Lokers, R. Knapen, S. Janssen, Y. van Randen, and J. Jansen. Analysis of Big Data technologies for use in agro-environmental science. *Environmental Modelling and Software*, 84 :494–504, 2016.

[LKZ16] C. Largouët, O. Krichen, and Y. Zhao. Temporal planning with extended timed automata. In *28th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2016*, pages 522–529, 2016.

[LLMP08] D. Lorenzoli, L. L. Mariani, and M. Pezzè. Automatic generation of software behavioral models. In *30th International Conference on Software Engineering*, pages 501–510, May 2008.

[LNGII19] T. Le Nguyen, S. Gsponer, I. Iulia, and G. Ifrim. Interpretable time series classification using all-subsequence learning and symbolic representations in time and frequency domains. *Data Mining and Knowledge Discovery*, pages 1–40, 2019.

[LPBBW13] C. Le Page, D. Bazile, N. Becu, and J. Weber. *Agent-Based Modelling and Simulation Applied to Environmental Management*, pages 499–540. Springer-Verlag, 2013.

[LPY97] K.G. Larsen, P. Pettersson, and W. Yi. Uppaal in a nutshell. *Journal of Software Tools for Technology Transfer*, 1(1-2) :134–152, 1997.

[LRG+18] I. Lage, S. Ross, S. J. Gershman, B Been Kim, and F. Doshi-Velez. Human-in-the-loop interpretability prior. In *31st Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 10180–10189, 2018.

[LZ03] G. Lamperti and M. Zanella. *Diagnosis of Active Systems*. Kluwer Academic Publishers, 2003.

[LZC14] C. Largouët, Y. Zhao, and M.-O. Cordier. Model-checking an ecosystem model for decision-aid. In *26th IEEE International Confe-*

rence on Tools with Artificial Intelligence, ICTAI 2014, pages 539–543, 2014.

[LZC16] C. Largouët, Y. Zhao, and M.-O. Cordier. Searching for cost-optimized strategies : An agricultural application. In *International Conference on Decision Support System Technology ICDSST 2016*, pages 31–43, 2016.

[LZVW19] Q. Lin, Y. Zhang, S. Verwer, and J. Wang. MOHA : A multi-mode hybrid automaton model for learning car-following behaviors. *IEEE Trans. Intelligent Transportation Systems*, 20(2) :790–796, 2019.

[MB14] R. Mathevet and F. Bousquet. *Résilience & Environnement. Penser les changements socio-écologiques*. Buchet - Chastel, 2014.

[MBK+19a] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor. Iot and agriculture data analysis for smart farm. *Computers and Electronics in Agriculture*, 156 :467–474, 2019.

[MBK+19b] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, and P. Wanichsombat, A.and Nillaor. IoT and agriculture data analysis for smart farm. *Computers and Electronics in Agriculture*, 156(June 2018) :467–474, 2019.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[MLTL05] M. Mangeas, C. Largouët, Touraïvane, and D. Lille. Le système EAUNET. un système d'information intégrant une modélisation dynamique des risques de pollution pour l'aide à la gestion de l'eau douce. *Ingénierie des Systèmes d'Information*, 10(4) :11–34, 2005.

[MMG10] S.J. Metcalf, K. Moyle, and D.J. Gaughan. Qualitative analysis of recreational fisher response and the ecosystem impacts of management strategies in a data-limited situation. *Fisheries Research*, 106(3) :289–297, 2010.

[MMH15] J. Merkert, M. Mueller, and M. Hubl. A survey of the application of machine learning in decision support systems. In *23rd European Conference on Information Systems, ECIS 2015*, 2015.

[MMPK17] S. Matthews, A. Miller, T. Plötz, and I. Kyriazakis. Automated tracking to measure behavioural changes in pigs for heath and welfare monitoring. *Scientific Report*, 7, 2017.

[MNA17] J. Majumdar, S. Naraseeyappa, and S. Ankalaki. Analysis of agriculture data using data mining techniques : application of big data. *Journal of Big Data*, 4(1) :20, 2017.

[Mur03] J. D. Murray. *Mathematical Biology II : Spatial Models and Biomedical Applications*, volume 18 of *Interdisciplinary Applied Mathematics*. Springer New York, 2003.

[NE89] J. Noblet and M. Etienne. Estimation of sow milk nutrient. *Journal of Animal Science*, 67(12) :3352–3359, 1989.

[NQH+12] T. Ngo, N. Quiniou, S. Heugebaert, F. Paboeuf, and J.-Y. Dourmad. Influence du rang de portée et du nombre de porcelets allaités sur la production laitière des truies. In *Journées Recherche Porcine*, pages 195–196, 2012.

[NS63] A. Newell and H. A. Simon. GPS, a program that simulates human thought. In E. A. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 279–293. McGraw-Hill, 1963.

[NS72] A. Newell and H. A. Simon. *Human Problem Solving*. Englewood Cliffs, 1972.

[OFCF11] A. Orlandini, A. Finzi, A. Cesta, and S. Fratini. Tga-based controllers for flexible plan execution. In *KI 2011*, pages 233–245, 2011.

[OPK+17] L. Olander, S. Polasky, J. Kagan, R. Johnston, L. Waigner, D. Saah, L. Maguire, J. Boyd, and D. Yoskowitz. So you want your research to be relevant ? building the bridge between ecosystem services research and practice. *Ecosystem Services*, 26 :170–182, 2017.

[Pai11] C. Paillette. Ecomata appliqué à un écosystème halieutique de la mer du nord. Master's thesis, Agrocampus Ouest, 2011. Rapport de stage de Master 2, Ingénieur Agronome, spécialité modélisation et écologie.

[Par87] D. Partridge. The scope and limitation of first generation experts systems. *Future Generation Computer Systems*, 3(1) :1–10, 1987.

[PCL+11] C. Paillette, G. Christansen, C. Largouët, L. Bonneau de Beaufort, and Fontenelle G. Modéisation qualitative de l'écosysème de la coquille saint-jacques en rade de brest avec ecomata. Technical report, Agrocampus Ouest, 2011. rapport de Projet Liteau.

[PCR02] Y. Pencole, M.-O. Cordier, and L. Roze. A decentralized model-based diagnostic tool for complex systems. *International Journal on Artificial Intelligence Tools*, 11(3) :327–346, 2002.

[Pea93] J. Pearl. From conditional oughts to qualitative decision theory. In *9th International Conference on Uncertainty in Artificial Intelligence*, pages 12–20, 1993.

[PEF00] H.-L. Pesonen, T. Ekvall, and G. et al. Fleischer. Framework for scenario development in lca. *The International Journal of Life Cycle Assessment*, 5(1) :21, Jan 2000.

[PFGL11] C. Paillette, G. Fontenelle, D. Gascuel, and C. Largouët. un outil de modélisation écosystémique qualitative : test en mer du nord. In *10 ème forum halieumétrique, Boulogne-sur-mer*, 2011.

[PG17] A. Panella and P. Gmytrasiewicz. A Survey of Multi-Agent Decision Making. *Auton Agent Multi-Agent Syst*, 31 :861–904, 2017.

[PI75] J. Pickands III. Statistical inference using extreme order statistics. *the Annals of Statistics*, pages 119–131, 1975.

[PK17] A. Patel and D. Kathiriya. Data Mining Trends in Agriculture : a Review. *AGRES - An International Journal*, 6(4) :637–645, 2017.

[PL85] C.J. Pucia and R. Levins. *Qualitative Modeling of Complex Systems : An Introduction to loop Analysis and Time Averaging*. Harvard University Press, 1985.

[PMM17] F. Pastore, D. Micucci, and M. Mariani. Timed k-tail : Automatic inference of timed automata. In *IEEE International Conference on Software Testing, Verification and Validation, (ICST)*, pages 401–411, 2017.

[Pom92] C. Pomerol. Autour du concept de décision assistée par ordinateur. *Journal of Decision Systems*, 1(1), 1992.

[Pom97] J.-C. Pomerol. Artificial intelligence and human decision making. *European Journal of Operational Research*, 99 :3–25, 1997.

[QAB+13] G. Quesnel, M. Akplogan, M. Bonneau, R. Martin-Clouaire, N. Peyrard, J.-P. Rellier, R. Sabbadin, and R. Trepos. Décision dans les agro-écosystèmes. *RFIA*, 27(4-5) :409–442, 2013.

[Qui16] C. Quinet. Herbert Simon et la rationalité. *Revue française d'économie*, 9(1) :133–181, 2016.

[RB93] B. Roy and D. Bouyssou. *Aide Multicritère à la Décision : Méthodes et Cas*. Paris, Economica, 1993.

[Rei87] R. Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32(1) :57–96, 1987.

[RG07] J. Rintanen and A. Grastien. Diagnosability testing with satisfiability algorithm. In *International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 532–537, 2007.

[RK17] M. Rußwurm and M. Körner. Multi-temporal land cover classification with long short-term memory neural networks. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42 :551–558, 2017.

[RMF+15] J. Reum, P. McDonald, B. Ferriss, D. Farrell, C. Hervey, and P. Levin. Qualitative network models in support of ecosystem approaches to bivalve aquaculture. *ICES Journal of Marine Science*, 72(8), 2015.

[RN10] S.J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010.

[Roy85] B. Roy. *Méthodologie multicritère d'aide à la décision*. Economica, Paris, 1985.

[Roy90] B. Roy. Decision-aid and decision-making. *European journal of Operational Research*, 45(2-3) :324–331, 1990.

[Roy93] B. Roy. Decision science or decision-aid science ? *European journal of Operational Research*, 66 :184–203, 1993.

[Rud18] C. Rudin. Please stop explaining black box models for high stakes decisions. In *Advances in Neural Information Processing Systems, NIPS'18, Workshop on Critiquing and Correcting Trends in Machine Learning*, 2018.

[RW87] P. Ramadge and W. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control and Optimisation*, 25(1) :206–230, 1987.

[Ryk89] J. Rykiel. Artificial intelligence and expert systems in ecology and natural resources management. *Ecological Modelling*, 46(1-2) :3–8, 1989.

[SAL07] J. Scopélitis, S. Andrefouët, and C. Largouët. Modelling coral reef habitat trajectories : Evaluation of an integrated timed automata and remote sensing approach. *Ecological Modelling*, 205 :59–80, 2007.

[Sav54] L.J. Savage. *The Foundations of Statistics*. Wiley, New York, 1954.

[SBA06] P. Salles, B. Bredeweg, and S. Araújo. Qualitative models about stream ecosystem recovery : Exploratory studies. *Ecological Modelling*, 194(1-3) :80–89, 2006.

[SCL+19] J. Sa, Y. Choi, H. Lee, Y. Chung, D. Park, and J. Cho. Fast Pig Detection with a Top-View Camera under Various Illumination Conditions. *Symmetry*, 11(2) :266, 2019.

[SFTL17] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouët. Anomaly detection in streams with extreme value theory. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*, pages 1067–1075, 2017.

[SFTL18] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouët. Are your data gathered ? In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18)*, pages 2210–2218, 2018.

[SG10] M. Sadik and L. Gruenwald. Dbod-ds : Distance based outlier detection for data streams. In *International Conference on Database and Expert Systems Applications*, pages 122–136, 2010.

[SGS18] K. Sobocinsky, C. Greene, and M. Schmidt. Using a qualitative model to explore the impacts of ecosystem and anthropogenic drivers upon declining marine survival in pacific salmon. *Environmental Conservation*, 45(3), 2018.

[Sil81] B. W. Silverman. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society. Series B (Methodological)*, 43(1) :97–99, 1981.

[SSL+95] R. Sampath, S. Sengupta, S. Lafortune, K. Sinnamohideen, and Teneketzis D. Diagnosability of discrete event systems. *IEEE Transactions Automatic Control*, 40 :1555–1575, 1995.

[SSL+96] R. Sampath, S. Sengupta, S. Lafortune, K. Sinnamohideen, and Teneketzis D. Failure diagnosis using discrete event models. *IEEE Transactions on Control Systems Technology*, 4(2) :105–124, 1996.

[STMLC14] A. Subias, L. Travé-Massuyès, and E. Le Corronc. Learning chronicles signing multiple scenario instances. *IFAC Proceedings Volumes*, 47(3) :10397–10402, 2014.

[TMID03] L. Travé-Massuyès, L. Ironi, and P. Dague. Mathematical foundations of qualitative reasoning. *AI Magazine, Special Issue on Qualitative Reasoning*, 24(4) :91–106, 2003.

[TN06] D. Tullos and M. Neumann. A qualitative model for characterizing effects of anthropogenic activities on benthic communities. *Ecological Modeling*, 196 :209–220, 2006.

[Tso08] A. Tsoukiàs. From decision theory to decision aiding methodology. *European Journal of Operational Research*, 187(1) :138–161, 2008.

[VNM47] J. Von Neumann and O Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[WdK89] D.S. Weld and J. de Kleer. *Readings in qualitative reasoning about physical systems*. Morgan Kaufmann Publishers Inc., 1989.

[WFHC17] H. Witten, E. Frank, M. Hall, and Pal C. *Data Mining, Practical Machine Learning Tools and Techniques*. Morgan Kauffmann, 4 edition, 2017.

[WWF74] P. Watzlawick, J.H. Weakland, and R. Fisch. *Change ; Principles of problem Formation and Problem Resolution*. Norton, New York, 1974.

[YZC11] C. Largouët Y. Zhao and M.-O. Cordier. Ecomata : Un logiciel d'aide à la décision pour améliorer la gestion des écosystèmes. *Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information*, 16(3) :85–111, 2011.

[Zad65] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338–353, 1965.

[ZCL12] Y. Zhao, M.-O. Cordier, and C. Largouët. Répondre aux questions "que faire pour" par synthèse de contrôleur sur des automates temporisés - application à la gestion de la pêche. In *18ème Congrès sur la Reconnaissance des Formes et Intelligence Artificielle (RFIA 2012)*, 2012.

[ZD08] X. Zhao and Ouyang D. Model-based diagnosis of discrete event systems with an incomplete system model. In *European Conference of Artificial Intelligence (ECAI'08)*, pages 189–193, 2008.

[Zha14] Y. Zhao. *Modélisation qualitative des agro-écosystèmes et aide à leur gestion par utilisation d'outils de model-checking.* PhD thesis, Université de Rennes 1, 2014.

[ZL13] J. Zaytoon and S. Lafortune. Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control*, 37 :308–320, 2013.

Curriculum Vitæ détaillé

Présentation générale

Nom :	Largouët
Prénom :	Christine
Adresse :	2 rue de viarmes 35000 Rennes
Nationalité :	Française
Situation de famille :	mariée, deux enfants
Grade :	Maître de conférences (informatique)
Etablissement :	AGROCAMPUS OUEST Unité Pédagogique Informatique 65 rue de St Brieuc - CS 84215 35042 Rennes Cedex
UMR :	IRISA, projet LACODAM Campus de Beaulieu 263 avenue du Général Leclerc 35042 Rennes Cedex
Téléphone :	06.33.61.94.55
Courrier électronique :	christine.largouet@irisa.fr

Parcours professionnel

2006-2019 **Maître de conférences en informatique**,
Agrocampus Ouest
(En disponibilité pour convenance personnelle : sept 2011 - sept 2013)
Responsable de l'unité pédagogique Informatique
UMR : projet LACODAM (ex DREAM jusqu'en 2015), IRISA Rennes

2003-2006 Maître de conférences en informatique, délégation
Université de la Nouvelle-Calédonie (UNC)
Chercheuse associée à l'US Espace, IRD de Nouméa

2001-2003 Maître de conférences en informatique
Agrocampus Rennes
Chercheuse associée à l'équipe DREAM, IRISA Rennes

1997-2001 Assistante d'Enseignement et de Recherche Contractuel (AERC)
Agrocampus Rennes (ENSAR)
Doctorante de l'équipe DREAM, IRISA Rennes

1993-1996 Ingénieure en informatique
Délégation Générale pour l'Armement
Bâtiment d'Essais et de Mesures "Monge", basé à Brest

Formation

Nov 2000 **Doctorat en informatique de l'Université de Rennes I**
Mention Très Honorable
Sujet : *Aide à l'interprétation d'une séquence d'images par la modélisation de l'évolution du système observé.*
Application à la reconnaissance de l'occupation du sol
Directrice de thèse : Marie-Odile Cordier

Juin 1996 **DEA Informatique**, IFSIC, Université de Rennes I
filière Intelligence Artificielle
Sujet : Elaboration d'un module pédagogique pour un tuteur intelligent

Juin 1993 **Diplôme d'ingénieur** des Arts et Métiers, Paris
spécialité informatique

Enseignement

Au sein de l'UP Informatique, responsabilités suivantes :

— Responsable de la formation en informatique dans les deux cursus ingénieur en agronomie et agroalimentaire du site de Rennes d'Agrocampus Ouest (AO) depuis 2006.
— Responsabilité et rédaction de tous les supports des enseignements suivants (présentés depuis 2003) :

Big-Data	45h eq. TD	M2 AO DataSciences	depuis 2017
Programmation scientifique en Python	37h eq. TD	M1 AO	depuis 2013
Gestion de données	37h eq. TD	M1 AO	depuis 2013
Tronc Commun : Bases de données et algorithmique, programmation R	56h eq. TD	L3 AO Agro	depuis 2013
Programmation Python	24h eq. TD	L3 AO AgroAlim	depuis 2018
Bases de données	24h eq. TD	L2 AO AgroAlim	depuis 2018
Intelligence Artificielle	56h eq. TD	M2 AO Géomatique	2013-2017
Introduction à la programmation	42h eq. TD	M1 Rennes2 Géomatique	2013-2017
Base de données	48h eq. TD	L3 AO Agro	2005-2011
Bases de données avancées	20h eq. TD	M1 AO Agro	2005-2011
Genie Logiciel	40h eq. TD	M1 AO Agro	2005-2011
Programmation Java	40h eq. TD	M1 AO Agro	2005-2011
Système d'exploitation Linux	40h eq. TD	M1 AO Agro	2005-2011
Introduction à l'IA	40h eq. TD	M1 AO Agro	2005-2011
Systèmes d'exploitation	48h eq. TD	L1 UNC	2003-2005
Programmation réseau	55h eq. TD	L2 UNC	2003-2005
Programmation JAVA	62h eq. TD	L2 UNC	2003-2005
Programmation JAVA avancée	53h eq. TD	L3 UNC	2003-2005

Participation aux instances de l'établissement

— Depuis 2013 : membre du COREGE (Comité de la Recherche d'Agrocampus Ouest). Représentante de l'UMR IRISA à Agrocampus Ouest.
— Depuis 2017 : membre du COPIL "révision des enseignements L3/M1" à Agrocampus Ouest (depuis 2011).
— De 2006 à 2011 : élue au conseil des enseignants, Agrocampus Ouest
— De 2006 à 2011 : suppléante au conseil de l'enseignement et de la vie étudiante, Agrocampus Ouest
— De 2009 à 2011 : membre de la commission de titularisation des maîtres de conférences, Agrocampus Ouest
— De 2006 à 2007 : membre de la commission pédagogique, Agrocampus Ouest

Recherche

Thématiques

- Intelligence Artificielle
- Aide à la décision
- Modélisation de systèmes dynamiques
- Systèmes à événements discrets
- Diagnostic
- Supervision

Co-encadrement de thèse

- 2017-2020 Raphaël Gauthier, à hauteur de 50% avec Jean-Yves Dourmad (INRA)
- 2016-2019 Alban Siffer, à hauteur de de 25% avec Alexandre Termier et Pierre-Alain Fouque (IRISA)
- 2010-2014 Yulong Zhao, à hauteur de 25% avec Marie-Odile Cordier (IRISA) et Chantal Gascuel (INRA)
- 2002-2005, Alban Grastien, à hauteur de 50% avec Marie-Odile Cordier (IRISA)

Encadrement de post-doctorats

- 2009, Yves-Marie Bozec, docteur en écologie marine (post-doc de 1 an), à hauteur de 100%,
  (Aujourd'hui chercheur en écologie marine, Université de Queensland, Australie).

Encadrement ou co-encadrement de stages de Master

- 2019, Nathan Dahmen, M1 Sup Telecom Paris, spécialité informatique
- 2019, Corentin Raphalen, M1 Université de Rennes 1 spécialité bioinformatique
- 2017, Mai Dao, M2 Université de Hanoi, spécialité informatique
- 2016, François Laferrière, M1 Université d'Angers, informatique
- 2015, Pierre Gaignon, M2 Agrocampus Ouest, spécialité statistiques
- 2015, Omar Krichen, M2 ENSI de Tunis, spécialité informatique
- 2011, Charlotte Paillette, M2 Agrocampus Ouest, spécialité modélisation en écologie
- 2010, Yulong Zhao, M2 Université de Rennes 1, spécialité informatique
- 2009, Jaouad Boufkri, M2 Université de Rennes 1, spécialité informatique
- 2005, Mounir Lekouara, M2 Ponts et Chaussées Paris, spécialité informatique
- 2004, Julie Scopelitis, M2 Agrocampus Ouest, spécialité halieutique
- 2004, Alban Grastien, M2 Université de Rennes 1, spécialité informatique
- 2002, Laetitia Rousseau, M1 Université de Rennes 1, spécialité informatique

Jurys de thèse

- 2015, Ying Xiao, Université de Loraine, *Crop sequence patterns in the French mainland, Exploring the crop sequence patterns at a large regional scale : A case from French mainland*,
- 2014, Yulong Zhao, Université de Rennes 1, *Modélisation qualitative des agro-*

écosystèmes et aide à leur gestion par utilisation d' outils de model-checking
— 2005, Alban Grastien, Université de Rennes 1, *Diagnostic déecentralisé et en-ligne de systèmes à événements discrets reconfigurables*

Comités de programme

— 2016-2019, Membre du comité de programme de la conférence APIA, Applications de l'Intelligence Artificielle
— 2014, Membre du comité de programme de l'atelier ANAEnv 2014, ANAlyse de données ENVironnementales
— de 2006 à 2010, Membre du comité de programme de l'atelier RTE 2010, Raisonnement sur le temps et l'espace

Implication dans les projets - collaborations

— **#DigitAg** Institut Convergences dédié à l'Agriculture Numérique financé le cadre des Investissements d'Avenir. Participation à l'axe de recherche "Analyse/fouille/modélisation". Projet en cours depuis 2017.

— **Partenariat Irisa-Rennes Métropole** Projet de modélisation d'un "jumeau numérique de Rennes" permettant de simuler et de prévoir les fonctionnements urbains. Projet en cours depuis 2018. Responsable de l'axe "modélisation écosystémique".

— **Rad2Brest** Programme LITEAU du Ministère de l'Ecologie et du Développement Durable, 2009-2011. Projet sur l'analyse du déroulement, dans le temps et l'espace, des principales activités humaines en mer côtière avec pour objectif la création d'un outil d'aide à la concertation. Responsable : Françoise Gourmelon, LETG Brest Geomer.

— **APPEAU** Progamme ADD (Agriculture et Développement Durable) de l'ANR, 2005-2008. Projet portant sur l'amélioration des prises de décision collective dans le cadre de la gestion de la ressource en eau à l'échelle de territoires (bassins versants, périmètres irrigués, etc.). Responsable : Delphine Burger-Leenhardt, INRA.

— **MOM** Projet de recherche du Ministère de l'Outre-Mer, 2005 sur l'étude de l'évolution du système récifo-lagonaire de Koné (Nouvelle-Calédonie) soumis à des pressions anthropiques croissantes. Responsable : Guy Fontenelle, IRD.

— **LITEAU II** Projet intitulé "Développement minier et gestion intégrée côtière en milieu corallien : appui méthodologique à la zone d'influence du projet Koniambo en Nouvelle-Calédonie", programme du Ministère de l'Ecologie et du Développement Durable, 2003-2005. Responsable : Marc Leopold, IRD.

Valorisation de la recherche

- Conférence invitée au séminaire AMURE “EcoMata : une approche par modélisation qualitative et temporelle pour la gestion des écosystèmes marins”, 15 septembre 2016
- Invitation au CSIRO d’Hobart, Australie 2010 par Rich Little pour présenter la modélisation qualitative des écosystèmes, visite de 10 jours, 2010
- Invitation au CSIRO de Canberra, Australie par Sylvie Thiébaux pour présenter les travaux sur le diagnostic, visite de 5 jours, 2004
- Logiciel EcoMata : outil d’aide à la décision pour la modélisation et l’exploration des écosystèmes marins (http ://people.irisa.fr/Christine.Largouet/ecoMata.html). Présentation d’EcoMata auprès des gestionnaires de pêche, Brest, Octobre 2013.
- Logiciel PaturMata : outil d’aide à la décision pour la gestion des paturages (http ://people.irisa.fr/Christine.Largouet/paturmata.html)

Relectures

- ECML - PKDD European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (depuis 2019)
- Natural Computing - An international Journal (Spinger)
- International Journal of Agricultural and Environmental Information Systems
- Conférence APIA (Applications de l’Intelligence Artificielle)
- Revue Hermès TSI (Technique et Sciences Informatiques)
- Revue Hermès RIA (Revue d’Intelligence Artificielle)

Publications scientifiques

Les rangs sont donnés lorsque connus. Pour certaines conférences, ils sont issus du site "Core Conference Portal" http ://portal.core.edu.au/conf-ranks/.

Revues avec comité de sélection

[1] Raphaël Gauthier, Christine Largouët, Charlotte Gaillard, Laetitia Cloutier, Frédéric Guay, Jean-Yves Dourmad, "Dynamic modeling of nutrient use and individual requirements of lactating sows", *Journal of Animal Science*, Vol 97(7), pp.2822-2836, 2019 (Rang A, IF 1.697)

[2] Christine Largouët, Omar Krichen, Yulong Zhao, "Extended Automata for Temporal Planning of Interacting Agents", *International Journal of Monitoring and Surveillance Technologies Research (IJMSTR)*, Vol 5(1), pp.30-48, 2017

[3] , "Use Of Timed Automata And Model-Checking To Explore Scenarios On Ecosystem Models", *Environmental Modelling and Software*, Vol 30, pp.123-138, 2011 (Rang A, IF 4.99 Clarivate Analytics)

[4] Yulong Zhao, Christine Largouët, Marie-Odile Cordier, "EcoMata : Un logiciel d'aide à la décision pour améliorer la gestion des écosystèmes", *Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information*, Vol 16 (3), pp.85-111, 2011

[5] Julie Scopélitis, Serge Andrefouët, Christine Largouët, "Modelling coral reef habitat trajectories : evaluation of an integrated timed automata and remote-sensing approach", *Ecological Modelling*, Volume 205, Vol 1-2, pp.59-80, 2007 (Rang A, IF 2.507 Clarivate Analytics)

[6] Morgan Mangeas, Christine Largouët, Touraïvane, Didier Lille, "Le système EAU-NET : un système d'information intégrant une modélisation dynamique des risques de pollution pour l'aide à la gestion de l'eau douce", numéro spécial "Systèmes d'information spatio-temporels", *Revue Ingéniérie des Systèmes d'Information*, Vol 10(4), pp.11-34, 2005.

[7] Christine Largouët, Marie-Odile Cordier, "Improving the Landcover Classification using Domain Knowledge", *AI Communication special issue on Environmental Sciences and Artificial Intelligence*, volume 14-1, pp.35-43, 2001. (Rang B, Core Journal Portal)

Congrès d'audience internationale avec actes et comité de sélection

[8] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, Christine Largouët, "Are your data data gathered ? The Folding Test of Unimodality", *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD18)*, pp.2210-2218, London,

United Kingdom, Août 2018 (rang A+)

[9] Alban Siffer, Pierre-Alain. Fouque, Alexandre Termier, Christine Largouët, "Anomaly Detection in Streams with Extreme Value Theory", *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD17)*, pp.1067-107, Halifax, Canada, Août 2017 (rang A+)

[10] Christine Largouët, Omar Krichen, Yulong Zhao. "Temporal Planning with extended Timed Automata", *28th International Conference on Tools with Artificial Intelligence (ICTAI 2016)*, pp.522-529, San José, United StatesNov 2016 (Rang B)

[11] Christine Largouët, Yulong Zhao, Marie-Odile Cordier. "Searching for Cost-Optimized Strategies : An Agricultural Application", *International Conference on Decision Support System Technology ICDSST 2016*, Springer Verlag, pp.31-43, Plymouth, United-Kingdom, Mai 2016

[12] Marie-Odile Cordier, Christine Largouët, Yulong Zhao, "Model-Checking an Ecosystem Model for Decision-Aid", *IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI 2014)*, pp.539-543, Limassol, Cyprus, Nov 2014 (Rang B)

[13] Christine Largouët, Marie-Odile Cordier, Guy Fontenelle, "Scenario templates to analyse qualitative ecosystem models", *International Congress on Modelling and Simulation (MODSIM'09)*, pp.2129-2135, Cairns, Australia, 2009

[14] Alban Grastien, Marie-Odile Cordier, Christine Largouët, "Automata Slicing for Diagnosing Discrete-Event Systems with Partially Ordered Observations", *Advances in Artificial Intelligence (AI*IA'05)*, pp. 270-281, Milan, Italy, Septembre 2005.

[15] Alban Grastien, Marie-Odile Cordier, Christine Largouët, "Incremental Diagnosis of Discrete-Event Systems", *DX'05 (International workshop on principles of Diagnosis)*, pp.119-124, Pacific Grove, California, USA, Juin 2005. (Rang B)

[16] Alban Grastien, Marie-Odile Cordier, Christine Largouët, "Incremental Diagnosis of Discrete-Event Systems", *International Join Conference in Artificial Intelligence IJCAI'05*, pp 1564-1565, Edinbourg, Août 2005. (Rang A+)

[17] Alban Grastien, Marie-Odile Cordier, Christine Largouët, "First Step Towards Incremental Diagnosis for Discrete-Event Systems", *AI'05, the Eighteenth Canadian Conference on Artificial Intelligence*, pp.170-181, Montréal, Mai 2005 (Rang B)

[18] Alban Grastien, Marie-Odile Cordier, Christine Largouët, "Extending decentralized discrete-event modelling to diagnose reconfigurable systems","Using model-checking techniques for diagnosing discrete-event systems" Carcassonne, Juin 2004 (Rang B)

[19] Marie-Odile Cordier, Alban Grastien, Christine Largouët, Yannick Pencolé, "Efficient trajectories computing exploiting inversibility properties", *DX'03 (International workshop on principles of Diagnosis)*, pp. 93-98, Washington, Juin 2003 (Rang B)

[20] Marie-Odile Cordier, Christine Largouët, "Using model-checking techniques for diagnosing discrete-event systems", *MoChArt'02 Model Checking and Artificial Intelligence*, Juillet 2002.

[21] Christine Largouët, Marie-Odile Cordier, "Adding Probabilities to timed automata to improve landcover classification", *Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU),Workshop Spatio-Temporal Reasoning and Geographic Information Systems*, Toulouse, Août 2001.

[22] Christine Largouët, Marie-Odile Cordier, "Timed Automata Model to Improve the Classification of a Sequence of Images", *ECAI'2000 (European Conference on Artificial Intelligence)*, Berlin, Allemagne, pp. 156-160, Août 2000. (Rang A)

[23] Christine Largouët, Marie-Odile Cordier, "Improving the Landcover Classification using Domain Knowledge", *ISPRS'2000 (XIXth Congress of the International Society for Photogrammetry and Remote Sensing)*, pp. 538-545, Amsterdam, Pays-Bas, Juillet 2000.

[24] Christine Largouët, Marie-Odile Cordier, "Combining observations and expectations : application to the refinement of an image sequence classification". *Workshop on Fusion of Domain Knowledge with Data for Decision Support, UAI'2000 (Uncertainty on Artificial Intelligence)* , Standford, USA, Juin 2000.

[25] Christine Largouët, Marie-Odile Cordier, "Improving the Landcover Classification using Domain Knowledge", *BESAI'2000 (Workshop on Binding Environmental Sciences and Artificial Intelligence)*, Berlin, Allemagne, Août 2000.

Chapitres d'ouvrage

[26] Christine Largouët, Marie-Odile Cordier, "Reconnaissance de l'occupation du sol à l'aide d'un automate temporisé", Ouvrage *Raisonnements sur l'espace et le temps : des modèles aux applications*, Traité IGAT, série Géomatique, pp. 317-332, Lavoisier, Paris, 2007.

[27] Jean-François Mari, Christine Largouët, "Modèles graphiques pour le raisonnement temporel et spatial", Ouvrage *Raisonnements sur l'espace et le temps : des modèles aux applications*, Traité IGAT, série Géomatique, pp. 249-271, Lavoisier, Paris, 2007.

Congrès d'audience nationale avec actes et comité de sélection

[28] Yulong Zhao, Marie-Odile Cordier, Christine Largouët, “Répondre aux questions ‘Que faire pourpar synthèse de contrôleur sur des automates temporisés - Application à la gestion de la pêche” *18ème Congrès sur la Reconnaissance des Formes et Intelligence Artificielle (RFIA 2012)*, Lyon, France, Janvier 2012

[29] Charlotte Paillette, Guy Fontenelle, Didier Gascuel, Christine Largouët, “Un outil de modélisation écosystémique qualitative : test en Mer du Nord”, *10ème forum halieumétrique “Changement environnementaux et dynamiques halieutiques”*, juillet 2011

[30] Christine Largouët, Marie-Odile Cordier, “Patrons de scénarios pour l’exploration qualitative d’un écosystème”, *17ème Congrès sur la Reconnaissance des Formes et Intelligence Artificielle (RFIA 2010)*, Caen, 20-22 janvier 2010

[31] Louis Bonneau de Beaufort, Guy Fontenelle, Christine Largouët, “OntoMap : ontologie et cartes cognitives”, *17ème Congrès sur la Reconnaissance des formes et intelligence artificielle (RFIA 2010)*, Caen, 20-22 janvier 2010

[32] Marie-Odile Cordier, Thomas Guyet, Christine Largouët, Véronique Masson, Henri-Maxime Suchier, “Apprentissage incrémental de règles de décision à partir de données d’un simulateur”, *Congrès Informatique des organisations et systèmes d’informations et de décision, atelier Systèmes d’Information et de Décision pour l’Environnement (INFORSID)*, Toulouse, 2009.

[33] Christine Largouët, Marie-Odile Cordier, “Modélisation par automate temporisé pour aider à l’identification de l’occupation du sol”, *12ème Congrès sur la Reconnaissance des Formes et Intelligence Artificielle (RFIA 2000)*, pp.280-294, Paris, 1-3 Janvier 2000.

Autres

[34] Raphaël Gauthier, Frédéric Guay, Ludovic Brossard, Christine Largouët, Jean-Yves Dourmad, “Precision feeding of lactacting sows : development of a decision support tool to handle variability”, *Annual Meeting of the European Federation of Animal Science*, Dubrovnik, Croatia, 5, Août 2018

[35] Christine Largouët, Yves-Marie Bozec, Yulong Zhao, “Integrating socio-economic drivers in an explicit-time, qualitative fisheries model : EcoMata”, *MSEAS 2016 Understanding marine socio-ecological systems : including the human dimension in Integrated Ecosystem Assessments*, Mai 2016 Brest, France

[36] Raphaël Gauthier, Frédéric Guay, Ludovic Brossard, Christine Largouët, Jean-Yves Dourmad,
“Développement d’un outil de prévision des besoins nutritionnels des truies en lacta-

tion. Application à l'alimentation de précision", Journées d'animation scientifiques du département Phase, April 2018

[37] Largouet, C., Fontenelle, G., Ferraris, J., Poignonec, "Les cartes cognitives : un outil pertinent pour initier un processus de gestion intégrée des zones côtières pour des écosystèmes lagoniares sous pression : une application en Nouvelle-Calédonie", *11th Pacific Science Intercongress*, Papeete, 2009.

[38] Scopélitis J., Andréfouët S., Largouet C., Phinn S. and Chabanet P. "Modelling and Mapping Coral Reef Habitats Transitions : an Integrated Remote Sensing and Artificial Intelligence Approach", GEOHAB 2007

[39] Marie-Odile Cordier, Thomas Guyet, Christine Largouët, Véronique Masson, Henri-Maxime Suchier, "Apprentissage incrémental de règles de décision à partir de données d'un simulateur", *Congrès Informatique des organisations et systèmes d'informations et de décision, atelier Systèmes d'Information et de Décision pour l'Environnement (INFORSID)*, Toulouse, 2009.

[40] Marie-Odile Cordier, Alban Grastien, Christine Largouët, Yannick Pencolé, "Calcul de trajectoires utilisant les propriétés d'interversibilité", *Rencontres jeunes chercheurs en Intelligence Artificielle (RJCIA'03)*, pp.15-28, Laval, 2003.

[41] Christine Largouët, Marie-Odile Cordier, "Interprétation d'une image à l'aide d'un modèle d'évolution du système observé. Application à la reconnaissance de l'occupation du sol, *conférence CARI'02 (6ème Colloque Africain sur la Recherche en Informatique*, Yaoundé, Octobre 2002.

[42] Christine Largouët, "Modélisation de l'évolution d'une parcelle agricole pour améliorer la reconnaissance de l'occupation du sol", *4ème Rencontres nationales des Jeunes Chercheurs en Intelligence Artificielle (RJCIA'98)*, pp.153-159. Toulouse, Septembre 1998.

Bilan des activités d'enseignement

Depuis septembre 2006 je suis responsable de l'Unité Pédagogique (UP) informatique du centre de Rennes d'Agrocampus Ouest. L'UP informatique est constituée de deux maîtres de conférences et d'un ingénieur de recherche. L'UP informatique appartient au département "Statistiques et Informatique". Le volume horaire des enseignements en informatique est d'environ 720 heures (équivalent TD) par année, réparti entre la filière d'ingénieur agronome, dite Agro (L3, M1), la filière d'ingénieur en agroalimentaire, dite AgroAlim, (L1, L2, L3, M1), le master Géomatique-Image (M1, M2) et le master Datasciences (M1,M2). Ces masters sont couplés à une spécialisation d'ingénieur agronome (M2).

Suite à des départs en retraite successifs en 2006, l'UP Informatique a connu une période de transition durant laquelle je fus la seule enseignante titulaire. À cette époque, le fonctionnement s'est alors appuyé sur du personnel contractuel renouvelé régulièrement. En 2009, un poste s'est ouvert et j'ai pu organiser un concours de maître de conférences qui a permis de recruter un titulaire. L'ingénieur de recherche du laboratoire a accepté de prendre en charge quelques enseignements ce qui a permis de stabiliser la situation. Depuis 2010, le volume d'enseignement de l'UP est répartie entre les deux enseignants-chercheurs, l'ingénieur de recherche et du personnel contractuel.

Responsabilités pédagogiques

Au sein de l'UP, mes responsabilités sont les suivantes :

— Proposer les programmes des enseignements d'informatique en cohérence à la fois avec les attentes de l'école (direction, autres disciplines), la CTI, et le profil des étudiants (non-informaticiens, issus de filières "sciences du vivant").
— Mettre en place les moyens nécessaires au bon déroulement des enseignements : répartition de la charge auprès du personnel et recrutements de maître de conférences contractuels ou de vacataires selon les années, possibilités techniques (en lien avec la DSI), organisation des effectifs dans les modules en fonction du type d'enseignement.
— Vérifier la qualité de l'enseignement réalisé, vis-à-vis des objectifs pédagogiques et de l'intérêt des étudiants. Apporter des éléments d'amélioration pour les années futures.
— Participer aux maquettes de Master (Géomatique, Datasciences) pour le volet informatique.

Modules enseignés

Mes enseignements concernent la formation initiale dans le cursus ingénieur Agro et AgroAlim et dans le master Géomatique-Image (jusqu'en 2017) et le master Datasciences

(depuis 2017) qui sont tous les deux cohabilités avec l'université de Rennes 2. Je suis aujourd'hui responsable des UC (Unité Constitutive) suivantes :

— Tronc Commun Informatique, L3 Agro
— Programmation scientifique, M1 Agro
— Gestion de données, M1 Agro,
— Base de données, L2 AgroAlim
— Programmation Python, L3 AgroAlim.
— Big Data, M2 Datasciences

Dans le master Géomatique-Image, j'ai enseigné l'UC "Introduction à la programmation" en M1, et l'UC "Introduction à l'Intelligence Artificielle" en M2. Depuis 2017 et la nouvelle maquette du master Datasciences, je donne un cours "Big Data" de 30h qui présente les concepts des traitements de gros volumes de données et les outils nécessaires à leur manipulation : Hadoop, MapReduce, Spark, Bases de données NoSQL : MongoDB, Neo4J.

Sans entrer dans le détail du contenu des différents cours qui restent assez classiques, je parlerais de l'UC "Informatique" des L3 Agro qui est l'enseignement le plus lourd puisqu'il concerne les 150 étudiants de L3 et représente pour certains élèves le seul enseignement en informatique de leur cursus. Suite à une réforme pédagogique, le programme de ce module a évolué récemment. Après concertation avec les enseignants des autres disciplines, il s'est avéré que deux thématiques étaient nécessaires à appréhender : les bases de données et la programmation. Le programme est ambitieux étant donné les effectifs et le volume horaire, trente heures par étudiant. Dans le module de SGBD, nous abordons une première partie assez théorique permettant la modélisation des données à l'aide d'un schéma relationnel. Cette approche permet aux étudiants de comprendre les bases de données sans les lier trop rapidement à un logiciel. Pour des aspects pratiques, nous avons opté pour le SGBD Access. Après diverses expériences par le passé (Base de OpenOffice, MySql), il s'avère que ce SGBD est le plus utilisé par nos collègues et requis par les entreprises du monde agronomique. Concernant la partie programmation, nous exposons dans un premier temps les concepts de l'algorithmique (variables, types, structures de contrôle, fonctions) puis les appliquons dans le langage de programmation R. Ce choix tout comme celui des bases de données permet de faire le lien avec les pratiques les plus utilisées dans les autres disciplines agronomiques. La difficulté de ce module provient des profils très variés des étudiants, certains venant de classe préparatoire et ayant déjà fait du Python, d'autres venant de filières "passerelles" et étant complètement novices. Il faut donc s'adapter aux différents niveaux et proposer des sujets modulables permettant de combler les attentes de chacun.

Pour tous les modules dont je suis responsable et présentés dans ce CV détaillé, je rédige la totalité des supports : supports de cours sous forme de diaporama, sujets de TD et de TP. Je rédige également les projets soumis en fin de module et qui changent d'une année sur l'autre, les sujets d'examen.

Ces enseignements sont complétés par des activités plus généralistes en agronomie (lorsque mes compétences suffisent), me permettant de m'insérer dans la vie de l'école. Ces activités sont le suivi de stages en exploitations agricoles et corrections des rapports (5 étudiants par an) ou les projets de l'enseignement "démarche scientifique". J'encadre également des groupes d'étudiants en M2 Datasciences, durant deux mois, sur des sujets d'initiation à la recherche et en lien avec des entreprises (pour cette année 2019, un groupe avec la startup Frigo Magic et un autre avec la société Givaudan).

Participation à l'enseignement hors établissement

Dans le cadre d'une délégation à l'Université de la Nouvelle-Calédonie (UNC) de 2003 à 2006, j'ai enseigné durant trois années scolaires l'informatique à des étudiants de la licence MISP (Mathématique, Informatique et Sciences Physiques). Durant cette période, j'ai eu la chance de participer à la mise en place du LMD, expérience très riche. En ce qui concerne la formation, j'ai enseigné : le système d'exploitation (Linux) en L1, les fondements de la programmation JAVA et réseaux en L2, et la programmation objet avancée en L3. J'ai apprécié de pouvoir approfondir mes enseignements grâce à des volumes horaires conséquents par module et à des étudiants motivés par la discipline. Ces trois années ont été un véritable enrichissement. Elles m'ont permis de comprendre le fonctionnement d'une université (par rapport à celui d'une école d'ingénieurs), de diversifier et de spécifier mes enseignements, de m'adapter à des étudiants différents des écoles d'ingénieurs et de la métropole (étudiants issus des communautés diverses de la Nouvelle-Calédonie).