

# SÉCURITÉ INFORMATIQUE

ACI « VERSYDIS »

## DESCRIPTION DU PROJET

### 1 – Objectifs et contexte

**Objectifs.** Le calcul concurrent et distribué est omniprésent. Les processeurs modernes exécutent plusieurs tâches en parallèle. La plupart des systèmes, comme les téléphones mobiles, les automobiles, les trains, sont construits à partir de modules qui communiquent entre eux. La même chose est vraie de très nombreux programmes et systèmes informatiques. La conception de tels systèmes, et la preuve de leur correction sont des tâches très difficiles et la moindre erreur peut avoir des conséquences extrêmement graves. Il est donc essentiel de concevoir des outils et des méthodes pour aider au développement de systèmes sûrs. Cela constitue un défi majeur et l'un des objectifs les plus importants de l'informatique fondamentale.

Le projet présenté ici concerne la sécurité des systèmes informatiques, et tout particulièrement des systèmes distribués décrits ci-dessus. Il s'agit de développer des concepts et des outils pour concevoir des systèmes sûrs et pour vérifier que le comportement de ces systèmes est conforme aux spécifications qui leur sont imposées. Plus précisément, ce projet s'inscrit dans le cadre de la vérification de modèle (model checking)[10, 6] et de la synthèse [49, 30, 9]. Une autre approche de la vérification s'appuie sur la déduction automatique, ce n'est pas la voie que nous suivrons.

**Contexte.** Ces dernières années, la vérification assistée par ordinateur a connu un développement spectaculaire, tant en ce qui concerne sa fondation théorique que ses applications. La méthode employée s'appuie sur les concepts de base de la théorie du calcul : les automates finis et la théorie des langages formels, de mots ou d'arbres. Le fondement de cette méthode est l'équivalence entre la notion algébrique de reconnaissabilité, la définissabilité logique et la reconnaissabilité par automates [56, 57, 47, 48]. Les deux premiers formalismes permettent de spécifier des propriétés des systèmes, et le troisième est utilisé dans les algorithmes de vérification – la vérification est ramenée au test du vide sur un automate.

Un problème voisin de la vérification est celui de la synthèse [49, 30, 9] : il s'agit de construire un système à partir d'une spécification donnée. L'avantage pour la sécurité est que le système obtenu par synthèse satisfait par construction sa spécification. L'étape de vérification a posteriori est donc superflue. Le problème de synthèse s'appuie sur les mêmes concepts, les mêmes résultats fondamentaux et les mêmes outils que celui de la vérification.

Les domaines fondamentaux décrits ci-dessus (automates, logiques, ...) bénéficient de plusieurs décennies de recherche. La synthèse et la vérification tirent ainsi profit de résultats théoriques puissants et de techniques algorithmiques efficaces. Cependant, ces techniques ne s'appliquent directement qu'aux systèmes séquentiels. Afin d'utiliser ces méthodes pour vérifier des systèmes distribués, ces derniers sont d'abord séquentialisés, ce qui engendre immédiatement une explosion du nombre d'états : la taille de la représentation des systèmes croît exponentiellement avec le nombre de leurs composants [8]. La difficulté à traiter des systèmes distribués déjà intrinsèquement très complexes se trouve ainsi multipliée. Pour donner un exemple très simple, supposons que nous avons  $n$  composants indépendants, chacun étant capable d'exécuter une unique action. N'importe laquelle des  $n!$  permutations de ces actions est un comportement possible, a priori différent des autres. Pourtant, connaissant la structure du système, nous savons que ces comportements devraient être considérés comme équivalents, c'est-à-dire qu'ils mèneront le système dans le même état quel que soit leur ordre d'exécution.

**Verrous.** La principale difficulté pour la vérification et la synthèse des systèmes distribués est d'éviter cette explosion du nombre d'états. De nombreux outils de vérification (model checkers) utilisent des heuristiques afin d'éviter de construire et d'explorer la totalité du système séquentialisé. Ces techniques sont connues sous le nom de méthodes de réduction d'ordres partiels [24, 25, 44, 45, 27]. Elles améliorent sensiblement l'applicabilité aux systèmes distribués des outils de vérification mais au fond, elles ne résolvent pas le problème de l'explosion du nombre d'états qui constitue depuis plusieurs années un verrou scientifique majeur.

**Projets similaires.** Les membres du projet participent à plusieurs projets sur des thèmes liés à celui-ci. Ainsi, à l'échelon français, ils animent avec d'autres une Action Spécifique du Département STIC du CNRS<sup>1</sup> et ce projet d'ACI peut être vu comme une première conséquence du travail de cette AS.

A l'échelon européen, nos équipes sont membres du projet RTN (Research and Training Network) GAMES, coordonné par E. Grädel (Aix-la-Chapelle), qui porte essentiellement sur l'utilisation de la théorie des jeux pour la vérification. Au-delà des frontières de l'Europe, un autre projet sur un thème

---

1. Cette AS, intitulée *Automates, modèles distribués et temporisés* et coordonnée par P. Weil, a débuté en octobre 2002.

proche, qui s'est révélé extrêmement productif est le projet franco-indien CEFIPRA<sup>2</sup> qui lie le LaBRI, le LIAFA et le LSV avec l'Institute of Mathematical Sciences et le Chennai Mathematical Institute.

La thématique de la vérification fait l'objet d'un foisonnement d'études dans le monde, et nous ne doutons pas qu'un bon nombre des réponses à cet appel d'offres porteront sur ce thème. Il ne nous est donc pas possible de citer tous les projets existants en France sur ce thème, encore moins de les commenter. Il nous semble cependant qu'ils seront peu nombreux à se baser sur l'exploitation de formalismes non séquentiels. Mentionnons cependant le projet *Persée* (méthodes symboliques pour la vérification des systèmes critiques hétérogènes), proposé par des (enseignants-)chercheurs du LSV, du LIAFA et du LaBRI en réponse à cette même ACI. Nous connaissons évidemment bien les porteurs de ce projet, membres des mêmes laboratoires que nous. Après en avoir discuté avec eux, nous avons conclu que nos objectifs concrets, et les outils et les techniques que nous souhaitons utiliser, sont trop différents pour justifier un regroupement qui ne soit pas que de forme.

## 2 – Description du projet

Afin d'aborder ce verrou de façon innovante, le projet se propose de rechercher des techniques de synthèse et de vérification qui exploitent l'aspect distribué des systèmes, i.e., utilisant le fait que la concurrence est déjà codée dans le modèle, pour obtenir de nouveaux algorithmes, plus efficaces que ceux basés sur l'entrelacement (interleaving) des actions. Autrement dit, travailler directement sur des modèles et des formalismes incorporant la concurrence doit permettre une meilleure compréhension des problèmes mais aussi d'obtenir de meilleurs algorithmes.

Un certain nombre de modèles de calculs, dits sans entrelacement, intègrent dans leur définition une notion de dépendance causale; c'est le cas des réseaux de Petri, de la dénotation de spécifications par diverses sortes de diagrammes ou par des automates à contrôle distribué (automates asynchrones ou automates communiquants) [7, 14, 60, 52]. Pour ces modèles, le problème de l'existence d'un grand nombre d'exécutions équivalentes ne se pose pas. En revanche, nous ne disposons pas, pour ces modèles plus complexes, de l'accumulation de résultats fondamentaux qui ont permis d'utiliser avec succès les modèles plus classiques mais moins puissants des arbres ou des mots à la vérification des systèmes séquentiels.

Notre projet vise à déterminer dans quelle mesure il est possible de procéder à synthèse et à la vérification directement sur les modèles sans entrelacement, en développant les résultats théoriques nécessaires et en explorant leur applicabilité. Nous souhaitons donc rester dans le monde sans entrelacement le plus longtemps possible. Pour cela, nous devons modéliser les systèmes par des formalismes sans entrelacement, utiliser des logiques et des automates adaptés à ces modèles, et finalement réduire le problème de la vérification à la résolution du problème du vide pour ces automates. L'espoir sous-jacent est que disposer d'une information explicite sur le parallélisme et la distribution peut aider dans la tâche de vérification.

De fait, cette approche est parfois tout à fait indispensable, par exemple lorsque les spécifications sont données par des diagrammes comme les HMSCs: dans ce cas, la vérification de propriétés exprimées en temps linéaire est indécidable, alors que la vérification de propriétés d'ordre partiel est décidable [36, 46]. De plus, utiliser un formalisme (modèle ou spécification) distribué a aussi l'avantage de mettre à jour des phénomènes que l'on ne peut pas observer ou décrire dans le cadre séquentiel. Par exemple, la propriété suivante d'un système réactif distribué ne peut pas s'exprimer avec les logiques arborescentes séquentielles: il existe un comportement distribué (le choix du comportement est fait par le système distribué) telle qu'une certaine propriété est vérifiée après l'exécution de n'importe quelle action initiale de ce comportement (l'environnement choisit l'action qui sera exécutée la première). Par contre, l'utilisation de logiques distribuées arborescentes permet de spécifier ce type de propriétés. Les propriétés de course (race conditions) dans les spécifications par HMSC [42], forment un autre type de propriétés importantes, qui concernent des questions de causalité, et qui s'expriment très difficilement ou pas du tout dans les modèles entrelacés.

Nous attaquerons également le problème de la synthèse à l'aide de modèles sans entrelacement. Cela pourrait nous éviter d'avoir à construire des produits de structures, et éliminer ainsi une des causes de l'explosion combinatoire du nombre des états.

Nous allons maintenant décrire plus en détail les modèles et les formalismes de spécification que nous allons étudier et utiliser, et préciser les problèmes concrets sur lesquels nous travaillerons.

### Modèles

Comme nous l'avons dit, nous allons étudier des modèles sans entrelacement de systèmes et de comportements. Des exemples classiques de modèles de systèmes, dans lesquels la notion de concurrence

---

2. Ce projet de trois ans, intitulé *Automata and concurrency: syntactic methods for verification* et co-dirigé par P. Weil et K. Lodaya, a débuté en août 2000.

et de causalité est explicite, sont les réseaux de Petri, les automates asynchrones, les HMSC (hierarchical message sequence charts) [14, 60, 52, 29, 28].

Pour rendre compte du comportement de ces modèles (réseaux, automates, HMSCs, etc), il est préférable d'utiliser des structures mathématiques qui intègrent elles aussi la notion de concurrence. Le plus simple de ces modèles, les traces de Mazurkiewicz [14], a été conçu précisément pour capturer la notion intuitive d'équivalence de comportements décrite dans l'introduction. Ce modèle a été beaucoup étudié, et on en connaît un grand nombre de propriétés. Les posets (ensembles partiellement ordonnés) constituent un modèle plus général [50]. On peut y avoir par exemple deux instances de la même action exécutées en parallèle. Les posets série-parallèles notamment ont été l'objet d'un certain nombre de travaux récents [32, 33, 34, 35]. Les MSC (message sequence charts) sont un autre exemple de modèle de posets qui a attiré beaucoup d'attention ces dernières années, notamment dans le contexte de la conception de protocoles de communication et en tant que norme de ITU [28].

Une trace ou un poset représente une exécution d'un système. Une structure d'événement au contraire est un modèle qui représente toutes les exécutions d'un système. Cette caractéristique est essentielle pour l'étude des propriétés arborescentes des systèmes distribués. L'étude des structures d'événements dans le contexte de la vérification et de la synthèse est relativement peu développée, et elle mérite incontestablement de l'être davantage.

## Formalismes de spécification

Le parti-pris de travailler directement avec des modèles sans entrelacement impose l'utilisation de formalismes adaptés pour décrire les propriétés des comportements de ces modèles. Comme dans le monde entrelacé, ces formalismes peuvent être logiques ou algébriques.

Observons, en ce qui concerne la logique, que les propriétés effectivement intéressantes à vérifier sur un modèle distribué sont celles qui sont elles-mêmes de nature distribuée. On connaît déjà quelques résultats d'indécidabilité, comme le problème de la vérification (model-checking) pour les HMSCs [5, 42], mais ces résultats sont en un sens peu pertinents pour les applications, car ils reposent sur une forme d'orthogonalité entre le modèle et la spécification.

Les logiques du premier ordre et du second ordre monadique s'étendent naturellement aux différents modèles que nous avons évoqués. Elles peuvent servir de référence à l'expressivité des autres formalismes logiques. La logique temporelle non branchante pour les traces a été l'objet de nombreux travaux ces dernières années [55, 4, 54, 15, 59, 22, 1]. En revanche, on sait peu de choses sur les logiques temporelles sur d'autres modèles (posets ou diagrammes par exemple) et il n'y a pratiquement aucun formalisme permettant de spécifier et de vérifier les propriétés arborescentes des systèmes distribués. L'approche algébrique s'applique relativement bien pour les traces [14] et dans une moindre mesure pour les ensembles de posets série-parallèle [33, 34], parce que l'on dispose d'un langage algébrique naturel et simple pour décrire ces objets.

Enfin, parce qu'ils constituent un formalisme concis et élémentaire, les automates peuvent aussi être utilisés pour spécifier le comportement souhaité d'un système. Les automates asynchrones pour les traces [61, 62, 21, 11, 13, 63] ou pour les pomsets [16, 31, 17] et les automates branchants pour les posets série-parallèles [34] sont des exemples intéressants d'extensions du modèle classique des automates finis.

## Jeux distribués

Les jeux peuvent être vus comme un modèle de calcul interactif, un formalisme de spécification et un outil technique. Dans le contexte des modèles sans entrelacement, les jeux servent de point de rencontre entre modèles et formalismes [57]. A l'un ou l'autre de ces titres, ils auraient pu être évoqués dans l'un des deux paragraphes précédents. Ils méritent une mention spéciale dans notre projet car les partenaires de ce projet ont développé ces toutes dernières années une compétence particulière dans ce domaine, au sein notamment d'un réseau européen.

Dans le cas séquentiel, la dernière étape de la vérification consiste à décider si le langage accepté par un certain automate est vide ou non. Une méthode générale pour arriver à cette décision est de résoudre un problème de jeu à deux joueurs naturellement associé [58]. Le problème de la synthèse de contrôleur se ramène également à la résolution d'un jeu.

Pour les modèles sans entrelacement, nous proposons de considérer des jeux dans lesquels l'environnement joue contre une coalition de joueurs, dont chacun n'a qu'une connaissance partielle de l'état global du jeu.

## Problèmes

Les paragraphes qui précèdent donnent déjà des pointeurs sur des domaines de recherche encore largement ouverts. Les questions plus précises listées ci-dessous figurent parmi celles qui nous semblent particulièrement importantes.

- Il n'est pas possible de résoudre algorithmiquement et de façon générale les jeux distribués. Nous voulons étudier les classes de jeux pour lesquels on peut déterminer algorithmiquement le gagnant et pour lesquels on peut calculer une stratégie gagnante. Nous voulons également étudier les opérations qui permettent de ramener l'étude d'un jeu distribué à l'étude d'un jeu plus simple (par exemple, un jeu ayant moins de joueurs).

- Nous utiliserons le modèle des jeux pour mieux comprendre le problème de la synthèse distribuée. Par exemple, il serait intéressant de trouver une formulation qui unifie les cas de décidabilité de la synthèse distribuée déjà connus [51, 49, 37, 38]. L'approfondissement de la théorie des jeux distribués peut aussi nous permettre d'en découvrir de nouveaux. Il peut enfin nous conduire à une meilleure compréhension de l'interaction entre modèles et spécifications, et à des résultats de décidabilité ou d'indécidabilité plus fins que ceux qui sont déjà connus, en considérant comme entrées du problème la paire constituée du modèle et de la propriété à vérifier.

- La notion de bisimulation [40, 43] joue un rôle important pour la spécification et la vérification des systèmes séquentiels. Les jeux fournissent un cadre idéal pour la définir et l'étudier. Il est dès lors très naturel d'étudier les liens entre les jeux distribués et les différentes bisimulations qui ont été spécifiquement développées pour les systèmes distribués (hereditary, history preserving, ...) [41].

- Nous souhaitons développer une méthode efficace pour synthétiser des automates asynchrones à partir d'une spécification de logique temporelle et caractériser la complexité du problème de synthèse distribuée. L'objectif final ici est de trouver de bons compromis entre expressivité et complexité.

- Les automates alternants sont utiles pour compiler de façon plus concise et plus rapide les formules de logique temporelle linéaire (LTL) [20]. D'autre part, l'alternance permet de représenter les choix (transitions non déterministes) d'un système devant se comporter correctement quelles que soient les réactions (transitions universelles) de l'environnement. Les choix possibles du système sont pris en compte dans la partie disjonctive de la fonction de transition des automates, localement sur chaque processus du système. Le comportement (espéré correct) du système face à toute réaction de l'environnement est pris en compte dans la partie conjonctive des transitions des processus modélisant l'environnement. Les automates asynchrones alternants n'ont pour l'instant pas été étudiés.

La question qui se pose naturellement est l'étude de ces automates et leurs connexions avec les jeux distribués (de même qu'il y a des connexions entre automates séquentiels et jeux).

- Pour la logique temporelle arborescente CTL sur les mots, l'algorithme de model-checking est linéaire par rapport à la taille de la formule, si le modèle est un automate séquentiel. En pratique on rencontre souvent des modèles distribués plus succincts, dont la synchronisation correspond aux automates asynchrones. Les logiques arborescentes, bien connues dans le cadre séquentiel, n'ont pas encore été transposées et étudiées dans le cadre distribué. Comme indiqué plus haut, elles sont essentielles pour étudier certaines propriétés des systèmes distribués réactifs.

- Certaines logiques temporelles sur les traces, les logiques locales [59, 22], ont la même complexité que LTL sur les mots (PSPACE-complet) dès que les opérateurs temporels qu'elles utilisent sont définis au moyen de formules de logique monadique du second ordre. Ces logiques laissent donc espérer, comme dans le cas des systèmes séquentiels, des algorithmes efficaces en pratique. Ceci justifie la recherche de logiques locales expressivement complètes. Certaines ont déjà été obtenues, mais elles utilisent des opérateurs de passé relativement peu aisés à manipuler. La recherche de logiques locales expressivement complètes, pur futur, et naturelles (permettant d'exprimer de façon intuitive des propriétés usuelles comme la sûreté, le blocage, la vivacité, etc) est donc un problème particulièrement important en pratique.

- Le dépliage de réseaux de Petri [39, 19] est une technique importante pour explorer l'espace des états d'un système sans pour autant vérifier chaque séquence d'exécutions possible. C'est un bon exemple d'une méthode où la connaissance de la structure du système aide à l'exploration des états. Pour le moment, cette méthode n'est utilisée que pour la vérification de problèmes de sûreté, en dehors de résultats encore préliminaires concernant les propriétés spécifiées en LTL [18]. Nous travaillerons à étendre cette méthode à la vérification de propriétés exprimées dans des formalismes sans entrelacement.

- Comprendre les connections entre les spécifications à base de diagrammes (HMSCs, LSCs [12], triggered MSCs [53]) et les modèles orientés machine comme les automates (communicants) est particulièrement important dans le cas des spécifications hétérogènes. On connaît déjà des travaux qui visent à traduire des HMSCs en automates communicants à états finis [2, 26, 3, 23]. De nombreuses questions restent à résoudre, comme de trouver un algorithme pour modifier les spécifications non exécutables. Nous pensons également utiliser des diagrammes pour contrôler l'interaction entre d'autres spécifications sans entrelacement comme les modules logiciels concurrents.

- Nous souhaitons explorer en détail l'extension des notions fondatrices de reconnaissabilité, régularité et rationalité, qui ont fait la preuve de leur pertinence pour les mots et les traces, au cadre plus général des ensembles de pomsets ou même de structures d'événements. Plus précisément, on s'intéressera par

exemple à la caractérisation algébrique et combinatoire des langages de posets série-parallèles définissables au premier ordre. Le problème de la définition d'un cadre algébrique permettant de traiter les langages de posets en général est très ouvert et mérite d'être étudié. Cela peut permettre de distinguer des classes de posets ou de structures d'événements ayant des propriétés plus agréables, et de les caractériser au moyen d'automates adéquats.

## Membres du projet et réseaux de recherche

Le projet regroupe des membres du LaBRI (David Janin (MC), Igor Walukiewicz (CR CNRS, HDR), Pascal Weil (DR CNRS)) et du LIAFA (Paul Gastin (PU), Anca Muscholl (PU), Marc Zeitoun (MC), Wieslaw Zielonka (PU)). Nous ne mentionnons qu'un nombre volontairement restreint de chercheurs qui ont déjà eu l'occasion de travailler ensemble. Les travaux de l'ACI s'appuieront, autant que de besoin, sur les autres compétences présentes dans nos laboratoires. De plus, nous associerons à ces travaux les étudiants et jeunes docteurs qui sont ou seront rattachés à nos équipes.

Il faut souligner que les directions de recherche qui sont développées dans ce projet représentent l'un des axes stratégiques et l'un des pôles d'excellence internationalement reconnus du LaBRI et du LIAFA.

Comme nous l'avons dit, les membres de ce projet se connaissent de longue date et ont une solide expérience du travail en commun dans ce domaine. On notera ainsi que Pascal Weil a été membre du LIAFA jusqu'en 1998, Wieslaw Zielonka a été membre du LaBRI jusqu'en 2002, et Anca Muscholl a fait un stage de 6 mois au LaBRI quelques années avant de devenir professeur au LIAFA.

Aujourd'hui, comme nous l'avons déjà mentionné, les membres de ce projet participent avec d'autres à une Action Spécifique du Département STIC du CNRS, à un projet RTN européen, et à un projet franco-indien CEFIPRA, tous sur des thèmes directement reliés au présent projet. De plus, nous nous appuyons, et continuerons à nous appuyer sur un réseau partagé d'autres collaborations internationales, moins formalisées. En particulier, un réseau de publications conjointes, d'échanges de visites, etc, établi de longue date, lie aussi nos équipes à celles des universités européennes de Aix-la-Chapelle (W. Thomas), Dresde (M. Droste, D. Kuske), Stuttgart (V. Diekert, J. Esparza), Varsovie (D. Niwinski), Wrocław (L. Pacholski).

**Valeur ajoutée** Pour permettre des avancées significatives sur les problèmes évoqués ci-dessus, il est essentiel d'utiliser des techniques avancées relevant de différents domaines. La collaboration entre les équipes concernées de Bordeaux et de Paris est à ce titre indispensable. Tous les membres du projet ont une assez bonne connaissance des divers domaines fondamentaux sur lesquels le projet se base. Cependant, nous ne pouvons individuellement prétendre à être spécialiste dans tous ces domaines. Au contraire, chacun est un spécialiste mondialement reconnu dans des domaines clés pour le projet :

- Le domaine d'expertise de Paul Gastin est la vérification des systèmes distribués. En particulier, c'est un spécialiste de la théorie des traces de Mazurkiewicz utilisées pour la modélisation des systèmes distribués ainsi que des logiques temporelles sur les ordres partiels utilisées pour la spécification des systèmes distribués. Il s'intéresse aussi aux automates et aux jeux distribués.
- Après avoir travaillé sur les fondements théoriques (en mathématique, logique et théorie des jeux) des langages de description de comportements tels que le mu-calcul, David Janin s'intéresse aujourd'hui aux problèmes de la spécification et de la synthèse de programmes distribués, en s'appuyant particulièrement sur la théorie des automates alternants et sur celle des jeux.
- Anca Muscholl a travaillé sur la théorie des traces de Mazurkiewicz puis s'est plus particulièrement orientée vers l'étude des langages de spécification à l'aide de diagrammes (MSC, HMSC, ...) dont elle est l'un des meilleurs spécialistes. Elle s'intéresse aussi à la théorie des automates et des jeux.
- Igor Walukiewicz est spécialiste du mu-calcul et des logiques pour décrire les comportements des systèmes. Il a aussi étudié ces logiques dans le contexte de la théorie des traces. Il est très largement reconnu pour l'application de la théorie des jeux aux problèmes de vérification et de synthèse.
- Pascal Weil est spécialiste de la théorie des automates finis et notamment de l'interaction entre ses aspects les plus algébriques (théorie des semigroupes finis) et ses aspects combinatoires et logiques. Il a aussi travaillé sur la reconnaissance de langages d'ensembles partiellement ordonnés, en particulier les langages série-parallèles, et de langages de graphes finis.
- Marc Zeitoun a étudié les propriétés syntaxiques de classes d'automates liées à des problèmes de décision. Récemment, il a travaillé sur des problèmes de vérification de langages de Message Sequence Charts (identification de classes expressives de MSCs ayant un model-checking efficace et pouvant être implémentés). Actuellement, il travaille particulièrement sur les automates d'ordres partiels et leur relation avec les jeux.
- Wieslaw Zielonka est spécialiste de la théorie des traces de Mazurkiewicz. Il a introduit les automates asynchrones, un modèle d'automates finis particulièrement bien adapté pour les systèmes distribués et qui joue pour les traces de Mazurkiewicz le même rôle fondamental que les automates

finis pour les systèmes séquentiels. Il travaille aussi dans le domaine de la théorie de jeux pour la modélisation de certains aspects du comportement de systèmes réactifs.

Le projet permettra une collaboration fructueuse qui tirera le plus grand profit de la mise en commun de ces diverses expertises.

### 3 – Résultats attendus

Étant donné qu'il s'agit d'un projet de recherche fondamentale, les résultats attendus sont principalement des publications dans des journaux et des conférences internationales. Notre action contribuera aussi à la formation de jeunes chercheurs (doctorants et post-doctorants). Enfin, nous diffuserons les connaissances élaborées durant le projet à l'occasion d'un workshop que nous souhaitons organiser en 2005.

Les questions que nous souhaitons aborder ont été suffisamment réfléchies afin qu'il n'y ait aucun risque d'absence de résultats. Cependant, nous ne sommes pas en mesure de donner un échéancier détaillé. Nous pouvons seulement indiquer le programme de travail précis que nous souhaitons adopter afin de favoriser la collaboration entre les deux équipes et de dynamiser les recherches sur les thèmes du projet. Nous prévoyons d'organiser une première rencontre de deux jours dès l'automne 2003, puis deux à trois rencontres par an. Ces rencontres permettront l'échange et la confrontation des points de vue, l'initiation de collaborations et la diffusion des résultats. Des visites individuelles régulières permettront un réel travail en commun entre membres du LaBRI et membres du LIAFA. Nous souhaitons inviter des chercheurs étrangers (seniors et juniors) spécialistes des domaines du projet. Cela augmentera l'impact international du projet que nous souhaitons résolument placer au meilleur niveau mondial.

L'impact potentiel du projet est d'ouvrir de nouvelles voies plus adaptées à la vérification des systèmes distribués. Ce sujet est l'un des plus importants pour la sécurité des systèmes informatiques. Notre approche innovante devrait déboucher sur de nouvelles techniques plus efficaces et sur la possibilité de prendre en compte des aspects de la vérification des systèmes distribués qui ne sont même pas exprimables avec l'approche entrelacée. Si nos espoirs se concrétisent, cela ouvrira la porte à la réalisation d'outils spécialisés dans la vérification des systèmes distribués.

Nous considérons que les risques scientifiques du projet sont minimes. Les objectifs que nous avons fixés sont réalistes. Tout au plus pourrait-on craindre que l'utilisation de formalismes non séquentiels ne réponde pas à toutes nos attentes en ce qui concerne l'amélioration de l'efficacité des techniques de vérification. Même si c'était le cas, il serait important de l'établir et l'effort de recherche consenti ne serait pas vain. De plus la possibilité de traiter de phénomènes liés à la sécurité des systèmes distribués et qui ne sont pas exprimables avec l'approche entrelacée constitue un enjeu qui suffirait à lui seul à justifier les recherches proposées dans ce projet.

## Références

- [1] B. Adsul and M. Sohoni. Complete and tractable local linear time temporal logics over traces. In *Proc. of ICALP'02*, number 2380 in Lecture Notes in Computer Science, pages 926–937. Springer Verlag, 2002.
- [2] R. Alur, K. Etessami, M. Yannakakis. Inference of message sequence charts. In *Proceedings of the 22nd International Conference on Software Engineering*, pages 304–313. ACM, 2000.
- [3] R. Alur, K. Etessami, M. Yannakakis. Realizability and Verification of MSC In *28th ICALP (F. Orejas, P.G. Spirakis, J. van Leeuwen, eds.)*, number 2076 in Lecture Notes in Computer Science, pages 797–808. Springer Verlag, 2001.
- [4] R. Alur, D. Peled, and W. Penczek. Model-checking of causality properties. In *Proc. of LICS'95*, pages 90–100, 1995.
- [5] R. Alur, M. Yannakakis. Model Checking of Message Sequence Charts. In *10th International Conference on Concurrency Theory CONCUR'99*, number 1664 in Lecture Notes in Computer Science, pages 114–129. Springer Verlag, 1999.
- [6] B. Bérard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, Ph. Schnoebelen. *Systems and Software Verification. Model-Checking Techniques and Tools*. Springer-Verlag, 2001.
- [7] D. Brand, P. Zafropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2):323–342, 1983.
- [8] J.R. Brunch, E.M. Clarke, K.L. McMillan, D.L. Dill, L.J. Hwang. Symbolic model checking:  $10^{20}$  states and beyond. *Information and Computation*, 98(2):142–170, 1992.
- [9] C.G. Cassandras, S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [10] E.M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 2000.

- [11] R. Cori, Y. Métivier, and W. Zielonka. Asynchronous mappings and asynchronous cellular automata. *Information and Computation*, 106:159–202, 1993.
- [12] W. Damm, D. Harel. LSCs: Breathing Life into Message Sequence Charts. *Formal Methods in System Design*, 19(1):45–80, 2001.
- [13] V. Diekert, A. Muscholl. Deterministic asynchronous automata for infinite traces. *Acta Informatica*, 31:379–397, 1994.
- [14] V. Diekert, G. Rozenberg, editors. *The Book of Traces*. World Scientific, 1995.
- [15] V. Diekert, P. Gastin. LTL is expressively complete for Mazurkiewicz traces. *Journal of Computer and System Sciences*, 64:396–418, 2002.
- [16] M. Droste, P. Gastin. Asynchronous cellular automata for pomsets without auto-concurrency. In U. Montanari and V. Sassone, editors, *Proceedings of the 7th International Conference on Concurrency Theory (CONCUR'96)*, number 1119 in Lecture Notes in Computer Science, pages 627–638. Springer Verlag, 1996.
- [17] M. Droste, P. Gastin, D. Kuske. Asynchronous cellular automata for pomsets. *Theoretical Computer Science*, 247:1–38, 2000.
- [18] J. Esparza, K. Heljanko. A new unfolding approach to LTL model checking. In *ICALP'00*, number 1853 in Lecture Notes in Computer Science, pages 475–486, 2000.
- [19] J. Esparza, S. Römer, W. Vogler. An improvement of McMillan's unfolding algorithm. *Formal Methods in System Design*, 20:285–310, 2002.
- [20] P. Gastin, D. Oddoux. Fast LTL to Büchi automata translation. In G. Berry, H. Comon, and A. Finkel, editors, *Proceedings of the 13th Conference on Computer Aided Verification (CAV'01)*, number 2102 in Lecture Notes in Computer Science, pages 53–65. Springer Verlag, 2001.
- [21] P. Gastin, A. Petit. Asynchronous cellular automata for infinite traces. In W. Kuich, editor, *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP'92)*, number 623 in Lecture Notes in Computer Science, pages 583–594. Springer Verlag, 1992.
- [22] P. Gastin, M. Mukund. An elementary expressively complete temporal logic for mazurkiewicz traces. In *ICALP'02*, volume 2380 of *Lecture Notes in Computer Science*, pages 938–949, 2002.
- [23] B. Genest, A. Muscholl, H. Seidl, M. Zeitoun. Infinite-State High-Level MSCs: Model-Checking and Realizability. In *29th ICALP (P. Widmayer, ed.)*, number 2380 in Lecture Notes in Computer Science, pages 657–668. Springer Verlag, 2002.
- [24] P. Wolper, P. Godefroid. Partial-order methods for temporal verification (invited paper). In *CONCUR'93*, volume 715 of *Lecture Notes in Computer Science*, pages 233–246. Springer-Verlag, 1993.
- [25] P. Godefroid, P. Wolper. A Partial Approach to Model Checking. *Information and Computation*, 110(2):305–326, 1994.
- [26] L. Hélouët, C. Jard. Conditions for synthesis of communicating automata from HMSCs. In *Proceedings of FMICS'2000, 5th Int. Workshop on Formal Methods for Industrial Critical Systems (S. Gnesi, ed.)*, 2000.
- [27] G.J. Holzmann. The model checker spin. *IEEE Trans. on Software Engineering*, 23(5):279–295, 1997.
- [28] *ITU-TS Recommendation Z.120: Message Sequence Chart (MSC)* ITU-TS, Geneva, September 1996.
- [29] K. Jensen. *Coloured Petri Nets, Vol. 1-3*. Springer-Verlag, 1997.
- [30] R. Kumar, V. K. Garg. *Modeling and control of logical discrete event systems*. Kluwer Academic Pub., 1995.
- [31] D. Kuske. Asynchronous cellular automata and asynchronous automata for pomsets. In *Proc. of CONCUR'98*, number 1466 in Lecture Notes in Computer Science, pages 517–532. Springer Verlag, 1998.
- [32] D. Kuske. A model theoretic proof of Büchi-type theorems and first-order logic for  $N$ -free pomsets. In *Proc. of STACS 2001*, number 2010 in Lecture Notes in Computer Science, pages 443–454. Springer Verlag, 2001.
- [33] D. Kuske. Towards a language theory for infinite  $N$ -free pomsets, *Theoretical Computer Science*, to appear.
- [34] K. Lodaya, P. Weil. Series-parallel languages and the bounded-width property. *Theoretical Computer Science*, 237:347–380, 2000.
- [35] K. Lodaya, P. Weil. Rationality in algebras with a series operation. *Information and Computation*, 171:269–293, 2001.
- [36] P. Madhusudan. Reasoning about Sequential and Branching Behaviours of Message Sequence Graphs. In *28th ICALP (F. Orejas, P.G. Spirakis, J. van Leeuwen, eds.)*, number 2076 in Lecture Notes in Computer Science, pages 809–820. Springer Verlag, 2001.

- [37] P. Madhusudan. *Control and Synthesis of Open Reactive Systems*. PhD thesis, University of Madras, 2001.
- [38] P. Madhusudan, P.S. Thiagarajan. A decidable class of asynchronous distributed controllers. In *CONCUR'02*, volume 2421 of *Lecture Notes in Computer Science*, 2002.
- [39] K.L. McMillan. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In *CAV'92*, volume 663, pages 164–174, 1992.
- [40] R. Milner. *Communication and Concurrency*. Prentice-Hall, London, 1989.
- [41] M. Mukund. Hereditary history preserving bisimulation is decidable for trace-labelled systems. In *FSTTCS'02*, *Lecture Notes in Computer Science*. Springer-Verlag, 2002.
- [42] A. Muscholl, D. Peled. Message sequence graphs and decision problems on Mazurkiewicz traces. In *24th Symposium on Mathematical Foundations of Computer Science (MFCS'99)* (M. Kutylowski, L. Pacholski, eds.), number 1672 in *Lecture Notes in Computer Science*, pages 81–91. Springer Verlag, 1999.
- [43] D. Park. Concurrency and automata on infinite sequences. In *Proc. of the 5th GI Conference on Theoretical Computer Science*, number 104 in *Lecture Notes in Computer Science*. Springer Verlag, 1981.
- [44] D. Peled. All from One, One for All: on Model Checking Using Representatives. In *Proceedings of Computer Aided Verification, 5th International Conference, CAV '93* (C. Courcoubetis, ed.), number 697 in *Lecture Notes in Computer Science*, pages 409–423. Springer Verlag, 1993.
- [45] D. Peled. Ten years of partial order reduction. In Alan J. Hu and Moshe Y. Vardi, editors, *Computer Aided Verification*, volume 1427 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.
- [46] D. Peled. Specification and verification of message sequence charts. In *Proceedings of FORTE/PSTV 2000* (T. Bolognesi, D. Latella, eds.), number 183 in *IFIP Conference Proceedings*, pages 139–154. Kluwer, 2000.
- [47] D. Perrin. Finite automata. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 1–53. North Holland, 1990.
- [48] D. Perrin, J.-E. Pin. *Infinite words*. Academic Press, 2003. to appear.
- [49] A. Pnueli, R. Rosner. Distributed reactive systems are hard to synthesize. In *31th IEEE Symposium Foundations of Computer Science (FOCS 1990)*, pages 746–757, 1990.
- [50] V.R. Pratt. Modelling concurrency with partial orders. *J. of Parallel Programming*, 15:33–71, 1986.
- [51] K. Rudie, W. Whonham. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. on Automat. Control*, 37(11):1692–1708, 1992.
- [52] V. Sassone, M. Nielsen, G. Winskel. Models for concurrency: Towards a classification. *Theoretical Computer Science*, 170(1-2):297–348, 1996.
- [53] B. Sengupta, R. Cleaveland. Triggered Message Sequence Charts. In *Proceedings of SIGSOFT2002/FSE-10*, pages 167–176. ACM Press, New York, 2002.
- [54] P. S. Thiagarajan, I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *LICS'97*, pages 183–194, 1997.
- [55] P.S. Thiagarajan. A trace based extension of linear time temporal logic. In *Proc. of LICS'94*, pages 438–447. IEEE Computer Society Press, 1994.
- [56] W. Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Vol.B*, pages 133–192. Elsevier, 1990.
- [57] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3. Springer-Verlag, 1997.
- [58] W. Thomas. Infinite games and verification. In *Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, pages 58–64. Springer-Verlag, 2002.
- [59] I. Walukiewicz. Local logics for traces. *Journal of Automata, Languages and Combinatorics*, 7(2):259–290, 2002.
- [60] G. Winskel, M. Nielsen. *Handbook of Logic in Computer Science*, volume 4, chapter Models for Concurrency, pages 1–148. Clarendon Press – Oxford, 1995.
- [61] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. — Informatique Théorique et Applications*, 21:99–135, 1987.
- [62] W. Zielonka. Safe executions of recognizable trace languages by asynchronous automata. In A. R. Meyer et al., editors, *Proceedings of the Symposium on Logical Foundations of Computer Science (Logic at Botik'89)*, number 363 in *Lecture Notes in Computer Science*, pages 278–289. Springer Verlag, 1989.
- [63] W. Zielonka. Asynchronous automata. In G. Rozenberg and V. Diekert, editors, *Book of Traces*, pages 175–217. World Scientific, Singapore, 1995.