

Outils pour représenter le temps dans la décision agronomique : Processus, Automates et logiques modales

Olivier Naud, Bertrand Léger, Florent Hernandez, Cemagref - UMR ITAP - TR CASYS
Version provisoire

Introduction

L'UMR ITAP participe à deux projets "Agriculture et Développement Durable" (ADD) et un projet « Programme National sur les Bio-énergies » (PNRB), pour lesquels le temps et les contraintes temporelles sont des éléments essentiels dans la représentation de la décision. Nos travaux utilisent notamment les Statecharts (Harel, 1987), les Automates Temporisés (Alur et Dill, 1990) et les logiques temporelles associées. Ils ont été présentés, sous différents aspects, à différentes communautés scientifiques (Léger et Naud, 2007 ; Léger et al, 2007 ; Naud et al, 2007A, 2007B). Nous allons ici évoquer les outils concrètement utilisés dans l'objectif d'échanger sur les méthodes et sur les savoir-faire avec les collègues.

Nous montrons l'intérêt des automates et autres formalismes de Systèmes à Événements Discrets (SED) dans une démarche de conception des processus de décision en production agricole. Nous assignons deux objectifs à cette conception. Le premier est d'analyser la performance attendue du système selon les critères du développement durable. Le second est de réaliser un transfert de connaissances et d'expertise au travers des modèles de processus de décision. Un tel transfert est une condition nécessaire à l'expérimentation à grande échelle de systèmes innovants complexes.

Projets et thèses liés à l'intervention

ADD GeDuQuE : « Innovations agro-écologiques et organisationnelles pour une Gestion Durable de la Qualité de l'Eau dans des régions de monoculture à forts niveaux d'intrants phytosanitaires »

<http://www.inra.fr/internet/Projets/add-vin/index.htm>

ADD Vin et Environnement : « Quelles interventions publiques et privées pour réduire l'utilisation des traitements phytosanitaires dans le secteur du vin ? »

<http://www.inra.fr/content/download/5367/52735/version/2/file/addprojet10-v2.pdf>

Thèse de Bertrand Léger : « conception, et formalisation par Systèmes à Événements Discrets, de processus de décision pour conduire une protection intégrée de la vigne »

<http://www.montpellier.cemagref.fr/ppp/olivier-naud/actu.html>

Thèse de Florent Hernandez : « Vérification et ordonnancement de systèmes temporisés... »

<http://www.montpellier.cemagref.fr/ppp/olivier-naud/TheseVerifOrdo.html>

Formalismes et sémantiques temporelles associées

Le formalisme des Automates à états fini est l'expression informatique fondamentale pour l'évaluation de la complexité d'un processus et le « model-checking » (vérification). Le temps est ici appréhendé à partir des événements (transitions entre états).

Les Statecharts (Harel, 1987) ont la même sémantique temporelle limitée que les automates. On peut exprimer des durées passées dans un état par le comptage d'événements (notamment les ticks d'une horloge fixe). Ce langage graphique est normalisé dans UML 2.0. Ils permettent une expression plus compacte que les automates à états finis grâce au double mécanisme de concurrence et d'imbrication. Le modèle est exécutable en simulation après compilation.

Les automates temporisés (Alur et Dill, 1990) sont des automates à états finis munis d'horloges en temps continu (dans R^+). Ces horloges permettent d'exprimer explicitement dates et durées comme contraintes temporelles sur les états et sur les transitions. Ces contraintes sont de la forme $x \# c$ ou $x - y \# c$, avec x et y deux horloges, c une constante (dans N) et $\#$ un symbole de relation de l'ensemble $\{<, \leq, =, \geq, >\}$. Les valeurs des horloges croissent uniformément dans le temps. Sur une transition, il est possible de remettre à zéro une ou plusieurs horloges, pour représenter des durées. La vérification de ces automates se fait grâce à une logique modale temporelle. Nous utilisons TCTL (Timed Computational Tree Logic), dont la vérification se fait en temps discret (c dans N) ou par régions symboliques de temps (vérifiant un jeu de contraintes temporelles). Dans la pratique, on dispose d'un système d'automates concurrents. La vérification s'effectue sur le produit, synchronisé par les événements, des automates. Avec un model-checker efficace, ce produit peut être limité aux parties nécessaires à la requête de vérification.

Simuler et vérifier

L'élicitation et la formalisation des connaissances ont un intérêt pour les experts-concepteurs de processus de décision et d'itinéraires techniques agro-environnementaux, qui trouvent dans le reflet qui leur est tendu l'occasion d'améliorer leur production. Au-delà de cet aspect cognitif, nous nous pencherons sur les possibilités offertes par la modélisation SED, et les outils associés, pour la simulation et la vérification.

Nous avons formalisé, sous le nom de « GrapeMilDeWS », un processus de décision pour la lutte combinée contre deux maladies de la vigne, le mildiou et l'oïdium. Ce modèle a été implémenté dans deux outils, Stateflow de Mathworks, et Rhapsody de I-logix. Le premier outil est attaché au progiciel de simulation Simulink, le second bénéficie du contexte objet de UML 2.0.

Le modèle « GrapeMilDeWS » reçoit en entrée des événements d'observations de niveau de maladie dans la parcelle de vigne considérée, ainsi que des prévisions climatiques. Il procure en sortie des événements correspondant aux décisions de traiter ou de pas traiter la parcelle. Il y a deux façons de simuler un modèle tel que celui-ci. La première est de réaliser un modèle détaillé des processus sous-jacents à la décision : développement de la vigne, développement des maladies, observation des maladies, mise en œuvre des décisions, ... La seconde est de fournir en entrée des fichiers d'événements codés en fonction du modèle du processus de décision. Ces fichiers constituent ce qu'on pourrait appeler des « scénarios ». C'est cette deuxième approche qui a jusqu'ici été privilégiée. Par comparaison du comportement du modèle vis à vis du comportement des experts prenant les décisions, nous avons pu vérifier la fidélité du modèle à l'expertise. Nous avons également pu mettre en évidence la gestion anticipée des contraintes temporelles que mettent en œuvre les experts-concepteurs à certains moments, imparfaitement prise en compte dans notre modèle Statecharts. En effet, dans un modèle SED déterministe, l'anticipation doit être représentée par des événements (prévisions météo par ex).

Avec un modèle à automates temporisés, en revanche, il est possible d'exprimer une non-détermination temporelle : telle transition sera franchie entre telle date et telle date. Sans rajout d'une procédure de résolution des contraintes temporelles, on ne peut pas simuler. On peut par contre vérifier que l'ensemble des comportements correspondant à un jeu de décision et de contraintes temporelles respecte certaines propriétés. La démarche est d'abord d'associer aux états des automates temporisés des propriétés logiques sous la forme de propositions atomiques, que l'on pourra retrouver dans la requête en vérification. La requête en vérification, en logique TCTL, détermine si, pour un état initial donné, on peut atteindre (il existe un chemin dans le graphe temporel), ou on atteint nécessairement (tous les chemins mènent à ...) un état final, souhaité ou craint. Du fait de ces opérateurs exprimant possibilité et nécessité, TCTL est une logique modale.

La présentation montrera comment utiliser Kronos (Yovine, 97) ou d'autres outils de model-checking, pour déterminer des dates au plus tôt et au plus tard pour des tâches avant de réaliser un ordonnancement.

Remerciements : Ce travail est mené en étroite partenariat avec l'équipe de protection intégrée de l'UMR Santé Végétale INRA-ENITAB (Léger et al, 2007). Nous leur exprimons toute notre reconnaissance.

Références

- Alur R., Dill D. L. (1990) Automata for modeling real-time systems. In *17th ICALP, Warwick University (England)*, 322–335. Voir aussi dans (1994) A theory of timed automata. *Theoretical Computer Science* **126-2**, 183-235
- Harel D. (1987) Statecharts : a visual formalism for complex systems. *Science of Computer Programming*, **8-3**, 231–274.
- Léger B., Cartolaro P., Delière L., Delbac L., Clerjeau M., Naud O. (2007) An expert based crop protection decision strategy against grapevine's powdery and downy mildews epidemics: Part 1) formalization. In: *Meeting of the IOBC/WPRS Working Group "Integrated Control in Viticulture", Marsala, Sicily, October 25-27*.
- Léger B., Naud O. (2007) Experimenting statecharts for multiple experts knowledge elicitation in agriculture. In: *IFAC-MCPL07 (Management and Control of Production and Logistics), Sibiu - Romania, 27-30 sept*, p. 869-874
- Naud O., Tuitete T., Léger B., Hélias A., Hernandez F., Giroudeau R. (2007A) Systèmes réactifs pour modéliser la décision en production agricole: Exemples à propos des traitements phytosanitaires. In: *MSR'2007 - Modélisation des Systèmes Réactifs, Lyon, France, 17-19 octobre*, p. 159-174.
- Naud O., Tuitete T., Léger B., Hélias A., Giroudeau R. (2007B) Systèmes à événements discrets : de la simulation à l'analyse temporelle de la décision en agriculture. In: *5^{ème} journées STIC & Environnement, Lyon, France, 13-15 novembre*.
- Yovine S. (1997) Kronos : A verification tool for real-time systems. (kronos user's manual release 2.2). *J^{al} of Software Tools for Technology Transfer* **1-(1-2)**, 123–133.