AGENCE NATIONALE DE LA RECHERCHE ANR	**Programme Jeunes chercheuses et jeunes chercheurs**	Réservé à l'organisme gestionnaire du programme N° de dossier : ANR-08-XXXX-00 Date de révision :
	Document scientifique associé	**Édition 2008**

Acronyme / *short title*	**HARRI**
Titre du projet *(en français)*	**H**iérarchisation et **A**pprentissage par **R**enforcement **R**elationnel **I**ndirect
Proposal title *(en anglais)*	*Hierarchical and Indirect Relational Reinforcement Learning*

Sommaire

1 Programme scientifique et technique / Description du projet 2
1.1 Problème posé / Abstract 2
1.2 Contexte et enjeux du projet 3
1.2.1 Apprentissage par Renforcement 3
1.2.2 AR Relationnel 4
1.2.3 ARR Indirect 6
1.2.4 Hiérarchisation et ARRI 7
1.2.5 Bibliographie 7
1.3 Objectifs et caractère ambitieux/novateur du projet 10
1.4 Description des travaux : programme scientifique 11
1.4.1 WP1 : Adaptation d'algorithmes classiques en AR au cas relationnel 11
1.4.2 WP2 : PLI pour une régression incrémentale efficace 12
1.4.3 WP3 : AR indirect dans le cadre relationnel 13
1.4.4 WP4 : Intégration de l'apprentissage de la fonction de transition et des fonctions valeur 15
1.4.5 WP5 : Exploitation de la fonction de transition pour l'AR hiérarchique 16
1.5 Résultats escomptés et retombées attendues 18
1.6 Organisation du projet 19
1.7 Organisation 21
1.7.1 Constitution de l'équipe proposée 21
1.7.2 Complémentarité et synergie des membres de l'équipe 21
1.7.3 Qualification du responsable scientifique et des membres de l'équipe : résumé et CV 21
1.7.4 Accès aux grands instruments 26
1.7.5 Fiche budgétaire 26
2 Justification scientifique des moyens demandés 27
2.1 Équipement 27
2.2 Personnel 27
2.3 Prestation de service externe 27
2.4 Missions 27
2.5 Dépenses justifiées sur une procédure de facturation interne 27
2.6 Autres dépenses de fonctionnement 27
Annexes 28
Description des participants 28
Biographies 28
Implication des personnes dans d'autres contrats 28

1 Programme scientifique et technique / Description du projet

1.1 Problème posé / Abstract

L'apprentissage par renforcement (AR) considère des systèmes informatiques engagés dans une boucle sensori-motrice (typiquement : un système robotique percevant son environnement par l'intermédiaire de capteurs, et doté de moyens d'action au moyen d'effecteurs). Plutôt que de programmer « à la main » les réactions du système dans chaque situation possible, on cherche à le voir acquérir automatiquement – par apprentissage – un comportement adéquat. Les techniques d'AR habituelles exploitent des représentations des états par attributs/valeurs comme par exemple un vecteur de distances aux objets les plus proches. La plupart des travaux en AR visent ainsi à découvrir des algorithmes d'apprentissage efficaces exploitant ces représentations propositionnelles. Dans le cadre du projet HARRI, nous entendons développer une activité en AR en ciblant nos recherches sur la question de la représentation des états et des actions. Avec des représentations utilisant des restrictions de la logique d'ordre un plutôt que des langages propositionnels, les situations sont représentées par des prédicats exprimant des relations entre objets dans l'environnement plutôt que par des vecteurs de valeurs numériques. Ce changement de paradigme offre de nouvelles possibilités d'apprentissage, notamment des possibilités de généralisation et de passage à l'échelle des solutions apprises qui sont hors de portée systèmes opérant dans des langages propositionnels. L'apprentissage par renforcement relationnel (ARR) mobilise ainsi des compétences complémentaires en :

- Apprentissage par renforcement (AR) pour découvrir de nouveaux algorithmes adaptés à de nouvelles représentations.
- Programmation logique inductive (PLI), un champ de l'apprentissage symbolique dédié à l'apprentissage de concepts ou de régularités exprimés en logique d'ordre un, pour découvrir de nouveaux algorithmes d'apprentissage relationnel adaptés au problème de l'apprentissage par renforcement (incrémentalité, stabilité).

L'équipe proposée pour le projet a ceci de particulier qu'elle dispose de jeunes spécialistes dans les deux domaines au sein du même laboratoire, ce qui est singulier en France. L'objectif scientifique du projet HARRI est d'investir rapidement un domaine émergent grâce à la complémentarité des participants. La part applicative prendra aussi une part importante des travaux engagés, avec des applications à des problèmes réels issus du domaine des jeux vidéo.

Reinforcement Learning considers systems involved in a sensor-motor loop (e.g., a robotic system perceiving its environment through sensors, and acting thanks to effectors). Instead of handcrafting the systems reactions in any possible situation, RL postulate is that the system should automatically acquire – through machine learning – adequate reactions. Classical RL techniques handle attribute value state representations, such vectors storing the distance of the robot with the nearest objects. Most research works in RL aims to discover efficient algorithms handling such propositional representations. In the HARRI proposal, we intend developing RL algorithms and specifically target algorithms that handle a more complex representation of states and actions. Assuming our target language for representing states and actions is a restriction of first order logic (refered hereafter as relational), situations are described in terms of predicates encoding relationships between objects of the environment, as opposed to numerical vectors for attribute-value representations. This paradigm shift opens new perspectives for reinforcement learning, in particular with respect to application scale-sup. Relational Reinforcement Learning (RRL) requires competence in both :

- *Reinforcement Learning for discovering new algorithms for making the most out of this new representation formalism*
- *Inductive Logic Programming, a subfield of symbolic Machine Learning dedicated to the development of algorithms that learns concepts and find regularities expressed in restrictions of First Order Logic, for discovering new algorithms that cope with the constraints of reinforcement learning, i.e., incrementality and stability.*

The team involved in the project gathers young researchers of both fields – RL and ILP – all working in the same research laboratory (note that LIPN is the only lab in France that gathers specialists of both fields). The scientific goal of the HARRI proposal is to quickly invest this rapidly emerging field of RRL, thanks to the complementary expertises of members of the group. Application development will also be an important aspect of the proposal, tackling real domains such as video-games.

1.2 Contexte et enjeux du projet

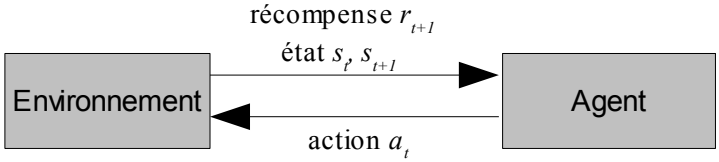
Un système automatique d'apprentissage par renforcement (Sutton & Barto, 1998) interagit avec son environnement et utilise l'information récoltée pour apprendre un comportement adéquat. Dans chaque situation qui peut se présenter, le système cherche à apprendre quelle action il vaut mieux entreprendre parmi celles qui sont à sa disposition. L'action la meilleure est celle qui devrait lui permettre de maximiser un cumul de récompenses obtenues sur le long terme. En effet, l'apprentissage automatique ne se fonde pas sur un oracle qui serait en mesure d'enseigner au système les meilleures actions dans chaque situation, mais plutôt sur une récompense scalaire sporadique, plus facile à définir par un concepteur que le détail des actions à entreprendre dans chaque situation possible. Fixer quand le système reçoit des récompenses et quand il n'en reçoit pas est un moyen indirect de lui assigner des buts. Le système cherche à maximiser le cumul de ces récompenses successives – sans connaître a priori son environnement – en effectuant un compromis entre récompenses immédiates et récompenses attendues dans le futur.

Les systèmes d'AR peuvent être envisagés partout où un système autonome doit s'améliorer au fur et à mesure de son expérience. Les domaines d'application les plus courants sont la robotique, les jeux vidéo ou la fabrication industrielle. Dans le cadre de ce projet, nous envisageons des applications au domaine des jeux vidéo, en plus des problèmes d'école usuels dans le domaine.

1.2.1 Apprentissage par Renforcement

En apprentissage par renforcement (AR), il est d'usage de formaliser les interactions entre un système et son environnement au moyen d'un Processus Décisionnel de Markov $<S,A,T,R>$ avec (Bellman, 1957) :

- S l'espace d'états ; les états successifs sont notés s_t, s_{t+1}...etc. Dans un jeu vidéo un état pourrait être la position de tous les opposants et la configuration du terrain, par exemple.
- A l'espace d'actions ; les actions choisies successivement sont notées a_t, a_{t+1}...etc. Une action pourraît être un déplacement ou un tir, par exemple.
- T la fonction de transition[1] $T:S\times A\rightarrow S$; s_{t+1} est l'état résultant du choix de l'action a_t dans l'état s_t. Une fonction de transition pourrait modéliser l'évolution de la position des opposants en fonction des déplacements du système joueur.
- R la fonction de récompense immédiate $R:S\times A\rightarrow \mathbb{R}$; r_{t+1} est la récompense immédiate reçue par le système après avoir entrepris l'action a_t dans l'état s_t. Une récompense négative pourrait sanctionner la mort du système, et une récompense positive pourrait être attribuée à la mort d'un ennemi.



Cette boucle sensori-motrice conduit donc le système à suivre une « trajectoire » $s_0, a_0, r_1, s_1, a_1, \ldots, r_T, s_T$. Au fil de ses interactions, il récolte des récompenses scalaires $r_1, r_2, r_3 \ldots r_T$.

L'objectif du système est d'apprendre une politique, c'est-à-dire comment agir de manière adéquate dans chaque état. Une politique est souvent représentée par une fonction $\pi:S\rightarrow A$ [2]. On dit qu'une politique est meilleure qu'une autre si elle permet de maximiser le cumul des récompenses obtenues au long terme. Ce cumul est souvent formalisé par un retour à compter d'un instant t donné égal à $\sum_{t=0}^{t=T}\gamma^t r_t$ avec $\gamma\in[0,1]$.

Selon que l'on recherche directement une politique ou non[3], on cherche souvent une approximation de la fonction valeur $V:S\rightarrow\mathbb{R}$ ou $Q:S\times A\rightarrow\mathbb{R}$ qui représente le retour attendu à partir du moment où

[1] Ici, on présente la version déterministe par souci de clarté. En principe, la fonction de transition est probabiliste : $T:S\times A\rightarrow\Pi(S)$

[2] Ici encore, on présente une version déterministe. En principe, la politique est probabiliste : $\pi:S\times A\rightarrow\Pi(S)$

[3] Dans ce document, nous parlerons essentiellement d'approches de l'AR fondées sur la valeur mais nos propositions concernant la question de la représentation restent appropriées pour des approches fondées sur la politique

l'on visite l'état ou le couple état/action en question. Selon le cas, dans un état s_t particulier, on préférera l'action a_t qui permet de parvenir dans l'état s_{t+1} en maximisant $r_{t+1} + V(s_{t+1})$ ou celle telle que $Q(s_t, a_t) = max_{a \in A} Q(s_t, a)$.

Une fonction valeur associe donc à chaque état – ou chaque couple état/action – un scalaire synthétisant l'espérance de toutes les récompenses futures. Un algorithme d'AR doit en conséquence permettre de « rétro-propager » les récompenses immédiates r_t obtenues vers les états et les actions décidées plus tôt et qui ont permis de l'obtenir finalement. En AR par différence temporelle TD (Sutton, 1988), par exemple, on cherche une approximation de cette fonction valeur optimale en ne conservant à chaque étape que le quadruplet $(s_t, a_t, r_{t+1}, s_{t+1})$ et le modèle courant de la fonction valeur.

Dès que la taille des problèmes augmente, il devient illusoire de vouloir apprendre la valeur chacune des multiples situations possibles. On représente alors de manière non-extensive les espaces d'états et d'actions continus ou de grande taille avec un nombre réduite de paramètres, et on exploite ces représentations dans des systèmes d'apprentissage (Bertsekas & Tsitsiklis, 1996). On parle alors d'approximation des fonctions valeur. Ces systèmes permettent de d'apprendre à quel point chaque action est adaptée dans chaque état, mais sans les énumérer tous. Toutes les techniques d'apprentissage supervisé sont susceptibles d'être utilisées en AR pour peu qu'elles soient incrémentales, puisque l'apprentissage opère au fil des interactions entre le système et son environnement.

1.2.2 AR Relationnel

Habituellement, l'AR utilise des représentations propositionnelles – par attributs/valeurs – pour produire des approximations de la fonction valeur. Si les états et les actions sont représentés par des vecteurs numériques, on peut par exemple employer des réseaux de neurones pour la trouver (Tesauro, 1995).

Dans le cadre de ce projet, nous nous intéressons aux représentations relationnelles pour l'AR. Dès lors, nous nous intéressons conjointement aux domaines de l'AR et de la programmation logique inductive (PLI). La PLI (Muggleton, 1991) est un domaine actif de l'apprentissage depuis les années 1990, qui se consacre à l'apprentissage dans des langages plus complexes que la logique attribut/valeur, le plus souvent des restrictions de la logique des prédicats. Le passage à un langage de concept plus complexe donne accès à des applications hors de portée des systèmes d'apprentissage manipulant des langages attributs/valeurs – toutes celles qui reposent sur la détection de régularités structurelles. En revanche, la plus grande complexité des langages cibles abordés s'accompagne d'une complexité accrue des processus d'apprentissage et d'exploitation des concepts appris. La PLI a rencontré de nombreux succès en apprentissage supervisé et non supervisé dans le domaine de l'extraction de connaissances dans des bases de données relationnelles (Dzeroski & Lavrac, 2001), mais s'est peu intéressé – jusqu'à récemment – à des contextes d'applications tels que l'apprentissage par renforcement, avec les contraintes d'incrémentalité et de stabilité de l'apprentissage.

Les représentations relationnelles pour l'AR ont été introduites par Dzeroski & al (2001) et suppose que les états et les actions soient représentés par des prédicats relationnels plutôt que par des vecteurs d'attributs. Tous les problèmes d'AR ne s'expriment pas d'emblée de manière relationnelle mais dans de nombreuses applications mettant en oeuvre des environnements simulés (jeux vidéo ou simulations aéronautiques par exemple) il peut être plus facile de produire une représentation relationnelle qu'une représentation propositionnelle. Dans le cadre du projet HARRI, nous entendons appliquer nos recherches au domaine des jeux vidéo.

Considérons pour l'instant un problème jouet habituel en ARR et qui consiste à résoudre un problème d'empilement dans un monde de cubes. Soit la configuration suivante, où certains cubes sont au sol et d'autres sont empilés :

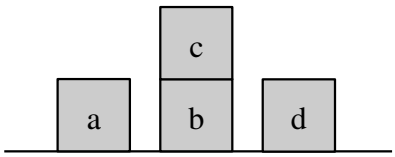


Fig 1 : Un état dans le monde des cubes

Dans le cadre habituel de l'AR, on représente les états et les actions par des vecteurs d'attributs. Ici, par exemple, l'état peut être représenté par la position de tous les blocs, chacun avec une abscisse et une ordonnée. Cet état s_t pourrait donc être représenté par les attributs suivants : ***(0**, 0, **1**, 0, **1**, 1, **2**, 0)*. En gras, nous avons indiqué l'abscisse de chacun des blocs a, b, c et d.

L'action a_t d'empiler le bloc *d* sur le bloc *c* pourrait être représentée par le vecteur *(2,1)*, pour signifier qu'on cherche à empiler le bloc en haut de la colonne 2 sur la colonne 1. L'état s_{t+1} résultant de cette action peut être représenté par les attributs ***(0**, 0, **1**, 0, **1**, 1, **2**, 0)* :

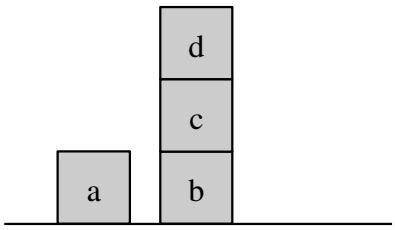


Fig 2 : Nouvel état après application de l'action Empiler (d,c)

Avec ce type de représentations propositionnelles, on pourra chercher à apprendre une politique qui permette de générer des règles de comportement comme ***(0**, 0, **1**, 0, **1**, 1, **2**, 0) → (2,1)*. On pourra également apprendre par régression incrémentale – en exploitant les valeurs des attributs – des approximations de la fonction valeur. Pour ce faire, on exploite la représentation des états et des actions, si bien que ce qui est appris est étroitement est lié au nombre de blocs utilisés. Ce qui a été appris avec un nombre de blocs donné est ainsi difficilement réutilisable avec un nombre de blocs plus important. Cette difficulté limite le passage à l'échelle par la réutilisation de solutions apprises dans un contexte simple, mais adaptées à des problèmes plus complexes.

Avec la possibilité d'un passage à l'échelle, il deviendrait envisageable de réutiliser des solutions apprises dans des versions simplifiées des problèmes pour amorcer l'apprentissage dans des cas plus complexes. Par exemple, pour traiter un problème à 10 blocs (58 941 091 états, trop complexe pour être traité d'emblée) on pourrait commencer par une version du problème à 5 blocs (501 états) et amorcer itérativement les apprentissages par les solutions obtenues à l'étape précédente :

$$\mathbf{Pb_{5\ blocs}} \xrightarrow{app_5} Sol_{5\ blocs} / \mathbf{Pb_{6blocs}} \xrightarrow{app_6} Sol_{6\ blocs} / \mathbf{Pb_{6\ blocs}} \xrightarrow{app_7} \ldots \mathbf{Pb_{10\ blocs}} \xrightarrow{app_{10}} Sol_{10\ blocs}$$

Pour aborder de manière naturelle ce passage à l'échelle, on peut se tourner vers des représentations relationnelles au lieu des habituelles représentations propositionnelles. Dans ce type de représentations, on utilise des objets (ici les blocs *a*, *b*, *c* et *d*) mis en relation par des prédicats relationnels (par exemple : *AuSol(bloc), Sur(bloc1,bloc2)...*). Avec une telle représentation, le premier état peut être représenté par la conjonction de prédicats suivante :

AuSol(a), AuSol(b), Sur(c,b), AuSol(d)

Les actions d'empilement peuvent également être représentées par des prédicats relationnels tels que *Empiler(d,c)* par exemple. Le second état peut être représenté par *AuSol(a), AuSol(b), Sur(c,b), Sur(d,c)*.

En exploitant ce type de représentation, on peut alors apprendre des règles de la forme *AuSol(a), AuSol(b), Sur(c,b), AuSol(d) → Empiler(d,c)*. Cette règle est utile si on cherche à empiler tous les blocs les uns sur les autres. Outre une plus grande lisibilité, on remarque que les objets utilisés dans l'action (c et d) dépendent des objets utilisés dans l'état. A une permutation près, une telle règle devrait pouvoir être généralisée à d'autres configurations d'empilement. En exploitant une représentation relationnelle, on peut trouver les nouvelles généralisations possibles correspondant à ces permutations en découvrant des règles d'ordre 1 comme :

AuSol(X), Sur(Y,Z) → Empiler(X,Y)

Ici, X, Y et Z sont des variables qui peuvent prendre n'importe quelle valeur. Cette règle d'ordre 1 est ainsi généralisable à toutes ses interprétations possibles. La valeur des variables dans la partie action est évidemment liée à celle dans les prémisses.

Outre une intelligibilité accrue et de nouvelles possibilités de généralisation, les représentations relationnelles offrent surtout de nouvelles possibilités pour le passage à l'échelle. En effet, une règle telle que la précédente ne préjuge pas du nombre de blocs du problème, et si elle avait été apprise dans une problème à 4 blocs, elle resterait utilisable telle quelle dans un problème à 5, 6 ou 8 blocs. Pour l'apprentissage d'une fonction valeur plutôt que directement d'une politique, on peut exploiter le même type de régularités entre états et actions au moyen de systèmes de régression incrémentale, pour peu qu'ils soient adaptés aux représentations relationnelles.

Il devient donc possible d'exploiter de nouveaux types de généralisation au moyen de systèmes de régression relationnelle. Le but est alors de tirer parti de la structure des représentations pour accélérer significativement la vitesse d'apprentissage, et améliorer l'intelligibilité du résultat de l'apprentissage.

A ce jour, la plupart des travaux en AR relationnel portent sur les systèmes de PLI pour la régression relationnelle employés, et peu encore explorent l'éventail d'algorithmes d'AR proprement dits. Les systèmes d'ARR utilisent en fait le plus souvent Q-learning pour la partie « AR » et proposent des améliorations dans la partie « PLI » avec des algorithmes comme TILDE-RT (Driessens & al, 2001), RIB (Driessens & al, 2003) ou KBR (Gärtner & al, 2003).

Le recherche en ARR peut pourtant suivre deux voies complémentaires :

- L'adaptation des algorithmes de programmation logique inductive aux contraintes spécifiques des problèmes d'AR, largement étudiée à l'heure actuelle ;
- L'adaptation des algorithmes classiques d'apprentissage par renforcement aux systèmes de PLI employés, pour l'instant plus confidentielle.

Nous proposons dans le projet HARRI d'exploiter nos compétences en AR pour explorer plus avant la seconde voie. Les recherches les plus récentes dans ce cadre restent à ce jour embryonnaires, si bien qu'un large domaine de recherche reste à investir au plus vite.

1.2.3 ARR Indirect

L'AR indirect est une des techniques d'AR les plus utilisées pour améliorer significativement la rapidité d'apprentissage. En AR direct, on cherche à exploiter l'expérience du système pour associer une valeur à chaque couple état/action possible, de manière à pouvoir choisir la meilleure action quand une situation se présente. Un tel système est donc en mesure de prédire la récompense qu'il attend des actions entreprises, mais jamais son nouvel état.

A cet apprentissage du retour attendu en terme de récompense, un AR indirect comme DynaQ ajoute celui d'un modèle prédictif des interactions entre le système et son environnement. Typiquement, un tel modèle peut consister en l'approximation de la fonction de transition $T:S\times A\rightarrow S$ prédisant le nouvel état du système en fonction de son dernier état et de l'action entreprise. Il permet d'anticiper toutes les conséquences des actions, et d'utiliser des techniques itératives de planification de manière à accélérer significativement l'apprentissage de la politique.

On peut aussi formuler le problème de l'AR indirect comme suit : au lieu d'utiliser l'interaction entre le système et son environnement pour apprendre directement une politique ou une fonction valeur V ou Q, on cherche en premier lieu à apprendre une approximation des fonctions $T:S\times A\rightarrow S$ et $R:S\times A\rightarrow \mathbb{R}$ qui sous-tendent les interactions entre le système et son environnement. Une fois les fonctions T et R connues – même imparfaitement – on peut exploiter cette connaissance pour calculer rapidement les fonctions $V:S\rightarrow \mathbb{R}$ ou $Q:S\times A\rightarrow \mathbb{R}$ avec des techniques de programmation dynamique (Bertsekas, 1995). Ce faisant, on n'apprend plus directement les fonctions valeur V et Q ou la politique : elle sont apprises indirectement, en employant comme intermédiaire les fonctions de récompense immédiate R et de transition T.

Avec ce type d'apprentissage, on utilise les modèles courants T_t et R_t de T et de R pour produire des exemples de tuples (s, a, r', s') obtenus en choisissant arbitrairement un couple (s,a) et en exploitant les modèles pour obtenir $s' = T_t(s)$ et $r' = R(s_t)$. Ces tuples construits – ou simulés – sont employés de la même manière que les nuplets $(s_t, a_t, r_{t+1}, s_{t+1})$ directement issus de l'expérience réelle, de manière à améliorer la politique et les fonctions valeur.

Pour procéder à un AR relationnel indirect, il est donc nécessaire d'être en mesure d'apprendre une approximation de :

- $R:S\times A\rightarrow \mathbb{R}$ par des techniques de régression incrémentale opérant sur des espaces d'états/actions relationnels ;
- $T:S\times A\rightarrow S$ par des techniques d'apprentissage incrémental permettant la prédiction d'ensembles de littéraux, explicitant le nouvel état ou bien les changements de l'état après application de l'action.

Les travaux les plus avancés en ARR indirect sont ceux de Croonenborghs et al (2007). Ces travaux récents restent toutefois assez limités dans la mesure où seuls quelques paramètres du modèle sont appris, et pas sa structure. Il reste ainsi beaucoup à faire et pour cela, nous emploierons des méthodes semblables

à celles développées par le porteur du projet mais dans le cadre des systèmes de classeurs à anticipation (Gérard et al, 2005), par décomposition de la fonction de transition.

Un des enjeux majeurs de notre projet sera donc de concevoir et de mettre en oeuvre une méthode d'AR relationnel indirect en utilisant les techniques de PLI pour apprendre également – incrémentalement – un modèle de l'environnement sous la forme d'un modèle partiel des actions. En en planification classique, les actions à apprendre étant exprimées dans des logiques d'ordre 1, nous développerons des techniques de PLI propres à apprendre ce type de modèles.

1.2.4 Hiérarchisation et ARRI

Parmi les autres succès récents les plus marquants en ARRI dans le domaine propositionnel, on peut citer l'AR hiérarchique au moyen de Semi-MDP (Boutilier et al, 1995). Un des travaux les plus spectaculaires dans cette approche en propositionnel, on peut citer Degris et al (2006). Le domaine applicatif choisi était aussi celui des jeux vidéo et plus particulièrement le jeu « Counter Strike » :

Roncagliolo & Tadepalli (2004) ont déjà adapté MAXQ au cadre relationnel. Cet algorithme opère par décomposition de la fonction valeur. Notre approche consistera plutôt à tirer parti de nos travaux en AR indirect en opérant par décomposition du modèle de l'environnement, à la manière de HEXQ (Hengst, 2002) dans le cadre propositionnel. Ces travaux nous mèneront en outre à travailler sur l'abstraction temporelle au moyen de hiérarchies, comme dans Bakker et Schmidhuber (2004).

Alors que les travaux sur les représentations relationnelles et le caractère indirect de l'AR sont bien définis et présentent un risque limité, ces travaux mêlant abstraction temporelle et hiérarchies pour l'AR indirect sont plus prospectifs.

1.2.5 Bibliographie

- Alphonse, E. (2003) *Macro-opérateurs et Sélection relationnelle en Programmation Logique Inductive : théorie et algorithmes*, Thèse de doctorat de l'Université de Paris 11
- Alphonse, E. and Rouveirol, C. (2006) *Extension of the Top-Down Data-Driven Strategy to ILP*. ILP 2006 : 49-63
- Asgharbeygi, N., Stracuzzi, D., and Langley, P. (2006) *Relational temporal difference learning.* Proceedings of the Twenty-Third International Conference on Machine Learning (pp. 49-56). Pittsburgh, PA.
- Bakker, B. and Schmidhuber, J. (2004) *Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization.* In A. Bonarini E. Yoshida F. Groen, N. Amato and B. Kr¨ose, editors, Proceedings of the 8-th Conference on Intelligent Autonomous Systems, IAS-8, Amsterdam, The Netherlands, pages 438–445, 2004.
- Bellman, R. E. (1957) *Dynamic Programming.* Princeton University Press.
- Bertsekas, D. P. (1995) *Dynamic Programming and Optimal Control.* Athena.
- Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neural Dynamic Programming.* Athena Scientific, Belmont, MA.
- Blockeel H. , De Raedt L. and Ramon J. (1998) *Top-down induction of clustering trees*. In Proceedings of the 15th International Conference on Machine Learning, pages 55--63.
- Boutilier, C., Dearden, R., and Goldszmidt, M. (1995). *Exploiting structure in policy construction.* In Proc. IJCAI, pp. 1104–1111.

- Boutilier, C., Reiter, R., & Price, B. (2001). *Symbolic dynamic programming for first order MDPs.* Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (pp. 690--697). Seattle, Washington.
- Breiman, L. , Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984) *Classification and regression trees*. Technical report, Wadsworth International, Monterey, CA.
- Bryant, C.H., Muggleton, S.H., Oliver, S.G., Kell, D.B, Reiser, P., and King R.D. (2001) *Combining inductive logic programming, active learning and robotics to discover the function of genes.* Electronic Transactions in Artificial Intelligence, 5-B1(012) :1-36.
- Croonenborghs, T., Ramon, J., Blockeel, H. and Bruynooghe, M. (2007) *Online Learning and Exploiting Relational Models in Reinforcement Learning.* IJCAI 2007 : 726-731
- Degris, T., Sigaud, O., and Wuillemin, P.-H. (2006) *Learning the Structure of Factored Markov Decision Processes in Reinforcement Learning Problems.* In Proceedings of the 23rd International Conference on Machine Learning (ICML), pages 257-264, Pittsburgh, Pennsylvania. ACM.
- Dietterich, T. G. (2000). *Hierarchical reinforcement learning with the MAXQ value function decomposition.* Journal of Artificial Intelligence Research, 13, 227--303.
- Dzeroski, S., De Raedt, L. and Driessens, K. (2001) *Relational reinforcement learning.* Machine Learning, 43 : 7-52.
- Dzeroski, S. and Lavrac N. (eds) (2001). Relational Data Mining, Springer, Berlin, 2001
- Driessens, K., Ramon, J., and Blockeel, H. (2001) *Speeding up relational reinforcement learning through the use of an incremental first order decision tree algorithm.* Proceedings of ECML - European Conference on Machine Learning (De Raedt, Luc and Flach, Peter, eds.), vol 2167, LNAI, pp. 97-108.
- Driessens, K. and Ramon, J. (2003) *Relational instance based regression for relational reinforcement learning.* Proceedings of the Twentieth International Conference on Machine Learning (Fawcett, T. and Mishra, N., eds.), pp. 123-130.
- Driessens, K. and Dzeroski, S. (2005) *Combining model-based and instance-based learning for first order regression*. Proceedings of the 22nd international conference on Machine Learning, pp 193 - 200
- Fern, A., Yoon, S., and Givan , R. (2006) *Approximate Policy Iteration with a Policy Language Bias : Solving Relational Markov Decision Processes* . Journal of Artificial Intelligence Research (JAIR), 25, 85-118, 2006
- Gärtner, T., Driessens, K., and Ramon, J. (2003) *Graph kernels and Gaussian processes for relational reinforcement learning.* Inductive Logic Programming, 13th International Conference, ILP 2003, Proceedings (Horvath, T. and Yamamoto, A., eds.), vol 2835, Lecture Notes in Computer Science, pp. 146-163, 2003
- Gérard, P. (2002) *Systèmes de classeurs : étude de l'apprentissage latent.* Thèse de Doctorat de l'Université Paris 6.
- Gérard, P., Meyer J.-A. and Sigaud, O. (2005) *Combining Latent Learning and Dynamic Programming in MACS.* European Journal of Operational Research 160 :614-637.
- Guestrin, C., Koller, D., Gearhart, C. and Kanodia, N. (2003). *Generalizing plans to new environments in relational MDPs.* IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence. Acapulco, Mexico : Morgan Kaufmann.
- Hengst, B. (2002) *Discovering hierarchy in reinforcement learning with HEXQ*. In C. Sammut and A. Hoffmann, editors, Proceedings of the International Conference on Machine Learning (ICML'02), Sidney, Australia, pages 243–250, San Francisco, 2002. Morgan Kaufmann Publishers Inc.
- Jaakkola, T., Jordan, M. I., and Singh, S. P. (1994). *On the convergence of stochastic iterative dynamic programming algorithms.* Neural Computation, 6.
- Moore, A. W. and Atkeson, C. G. (1993). *Prioritized sweeping : Reinforcement learning with less data and less real time.* Machine Learning, 13 :103-130.

- Muggleton, S.H. (1991) *Inductive Logic Programming.* New Generation Computing, 8(4) :295-318, 1991.
- Pasula, H.M., Zettlemoyer, L. S. , Kaelbling L. (2004) *Learning Probabilistic Relational Planning Rules.* In Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS-04), pp 73-82.
- Peng, J. and Williams, R. J. (1993). *Efficient learning and planning within the Dyna framework.* Adaptive Behavior, 1(4).
- Rodrigues, C. (2007) Traces d'éligibilité pour l'Apprentissage par Renforcement Relationnel. Thèse Master, Université Paris 13.
- Roncagliolo S. and Tadepalli, P. (2004) *Function Approximation in Hierarchical Relational Reinforcement Learning.* Workshop on Relational Reinforcement Learning, in conjunction with International Conference on Machine Learning 2004, Banff, Alberta, Canada
- Sutton, R. S. (1988). *Learning to predict by the method of temporal differences.* Machine Learning, 3 :9-44.
- Sutton, R. S. (1990). *Integrated architectures for learning, planning, and reacting based on approximating dynamic programming.* In Proceedings of the Seventh International Conference on Machine Learning, pages 216-224, San Mateo, CA. Morgan Kaufmann.
- Sutton R. S. and Barto A. G. (1998) *Reinforcement Learning : An Introduction.* MIT Press.
- Tadepalli, P., Givan, R., & Driessens, K. (2004) *Relational reinforcement learning : An overview.* Proceedings of the ICML'04 Workshop on Relational Reinforcement Learning (pp. 1--9). Banff, Alberta.
- Tesauro, G. J. (1995). *Temporal difference learning and TD-Gammon.* Communications of the ACM, 38 :58-68.
- van Otterlo, M. (2005) *A Survey of Reinforcement Learning in Relational Domains.* CTIT Technical Report series TR-CTIT-05-31 Centre for Telematics and Information Technology, University of Twente, Enschede. ISSN 1381-3625
- Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards.* PhD thesis, Cambridge University, England.
- Watkins, C. J. C. H. and Dayan, P. (1992). *Q-learning.* Machine Learning, 8 :279-292.

1.3 Objectifs et caractère ambitieux/novateur du projet

Habituellement, l'apprentissage par renforcement aborde le problème de l'adaptation d'un système autonome à son environnement d'un point de vue propositionnel. Les perceptions et les actions d'un tel système sont alors décrites comme des vecteurs d'attributs. Or, de nombreux problèmes sont facilement exprimables par des prédicats relationnels, explicitant les relations liant les objets de l'environnement. Lorsque c'est le cas, il semble opportun de chercher à exploiter la richesse de ces représentations pour améliorer la performance des algorithmes d'AR, et profiter de possibilités concernant le passage à l'échelle ou l'intelligibilité des solutions apprises.

Par ailleurs, le domaine de la PLI aborde principalement des problèmes d'extraction de connaissances, où il convient de produire automatiquement des modèles représentant des données issues d'une base de données. Sauf cas particuliers, les algorithmes de PLI peuvent souvent se passer de présenter certaines propriétés comme l'incrémentalité. Or dès que l'on prétend aborder des problèmes d'AR, une telle propriété devient absolument nécessaire. Ainsi, employer des algorithmes de PLI pour l'AR fait apparaître de nouvelles contraintes auxquelles ils doivent satisfaire, portant par exemple sur l'incrémentalité ou la stabilité de l'apprentissage. Du point de vue de l'apprentissage relationnel, l'ARR fait donc apparaître de nouvelles directions de recherche encore très partiellement explorées.

Le projet HARRI que nous proposons est donc novateur à plus d'un titre :

- Il ouvre de nouvelles perspectives de recherche en apprentissage par renforcement. Cette nouvelle direction est radicalement novatrice puisqu'elle porte sur les représentations, et que pour parvenir au même degré de maturité que l'AR propositionnel, tout reste à faire. En outre, le changement de représentations permettra d'aborder des problèmes pour l'instant hors de portée de l'AR propositionnel.
- Il pose de nouveaux problèmes auxquels les spécialistes de PLI ont moins l'habitude de devoir répondre (incrémentalité, contraintes sur la stabilité...). Là encore, dans un autre domaine, les domaines à explorer restent à ce jour assez vastes.
- Il fait travailler conjointement des communautés qui autrement sont assez disjointes.

Ce projet aborde donc un domaine émergeant de manière novatrice. On peut donc raisonnablement en attendre des progrès significatifs mais tout à fait à la portée d'un travail à moyen terme. Son cadre scientifique est clairement identifié et ses objectifs précis.

En outre, nous nous attacherons à valider expérimentalement nos algorithmes sur des problèmes réels issus du domaine applicatif des jeux vidéo. Ces expérimentations se feront dans le cadre de la politique de l'équipe consistant à capitaliser les différents travaux de recherche dans une plateforme logicielle fédératrice.

Rappelons également qu'il existe peu d'équipes dans le monde travaillant sur ce domaine et aucune en France pour l'instant.

1.4 Description des travaux : programme scientifique

1.4.1 WP1 : Adaptation d'algorithmes classiques en AR au cas relationnel

1.4.1.a Objectif scientifique

Les premiers travaux exploitant des représentations relationnelles en AR ont été menés à l'Université de Catholique de Leuven (KUL). Leurs travaux ont consisté à utiliser le couplage de Q-Learning avec plusieurs algorithmes de régression relationnels successifs : Tilde-RT, TG, RIB puis KBR.

Nos premier travaux dans le cadre de ce projet consisteront à emprunter la voie symétrique, à savoir utiliser les algorithmes de PLI développés à la KUL mais en cherchant à exploiter des algorithmes d'AR plus sophistiqués. En premier lieu, nous adapterons TD(λ) au cadre relationnel. Cette étape nous permettra de développer une plateforme expérimentale et de dépasser l'état de l'art. Certains travaux (Asgharbeygi & al, 2006) traitent déjà ce problème mais ils évitent les points difficiles en supposant une connaissance de la fonction de transition et ne cherchent à apprendre que des fonctions valeurs qui ne prennent pas en considération l'action. Il nous paraît abusif dans ce cas de parler de TD(λ).

1.4.1.b Programme

Dans un premier temps, nous envisageons de porter l'état de l'art en ARR (PLI + AR) dans la plateforme de PLI développée au sein du laboratoire par Erick Alphonse et Dominique Bouthinon. Cette plateforme inclut dans sa version actuelle des algorithmes d'apprentissage supervisé relationnel fondés sur des techniques de propositionnalisation contenant entre autres le système PROPAL (Alphonse et Rouveirol, 2006) qui permet un apprentissage relationnel efficace. Ainsi, cette étape consistera essentiellement à intégrer l'ARR dans une plateforme logicielle éprouvée et maîtrisée localement, de manière à faciliter les expérimentations ultérieures, et en particulier l'adaptation de TD(λ).

Nos premiers travaux sur le sujet sont déjà en cours et ont fait l'objet d'un stage de Master en 2007 (Rodrigues, 2007). Ils nous ont permis d'entamer une collaboration avec l'Université Catholique de Leuven. La KUL nous a aimablement fourni un accès au code source de tous leurs algorithmes d'ARR[4], si bien que nous avons pu démarrer cette activité dans de bonnes conditions.

Le WP1 se déroulera ainsi en trois étapes :

- Reproduction de l'état de l'art en régression incrémentale et intégration de Q-Learning sur la plateforme d'apprentissage relationnel du LIPN incluant PROPAL (4 mois) : le produit de cette étape sera une plateforme d'expérimentation. Cette étape suppose l'implémentation de plusieurs systèmes de PLI, avec la possibilité d'en changer facilement.
- Adaptation de TD(λ) aux représentations relationnelles (4 mois) : ces travaux seront sanctionnés par une publication sur le sujet, avec des expérimentations sur un problème jouet comme celui des mondes de blocs.
- Implémentation d'autres problèmes courants de l'état de l'art (4 mois), et étude expérimentale de TD(λ) dans plusieurs environnements (régression incrémentale, ...).

Le produit du WP1 sera donc :

- État de l'art en apprentissage par renforcement relationnel ;
- Une plateforme logicielle d'apprentissage par renforcement relationnel intégrée à la plateforme d'apprentissage relationnel du LIPN : système de régression relationnelle incrémentale
- Le dépassement de l'état de l'art grâce à l'adaptation de TD(λ) à une représentation relationnelle ;
- L'intégration de la plupart des problèmes utilisés en ARR dans la plateforme expérimentale. Les premiers de ces problèmes pourront par exemple être issus du General Game Playing Project[5].

Ces travaux nécessiteront la participation de

- Pierre Gérard, responsable, porteur et initiateur du projet ;

[4] `http ://www.cs.kuleuven.ac.be/~dtai/ACE`

[5] `http ://games.stanford.edu`

- Dominique Bouthinon et Erick Alphonse pour leur maîtrise de la plateforme d'apprentissage relationnel du LIPN et du système PROPAL ;
- Un ingénieur de recherche.

1.4.2 WP2 : PLI pour une régression incrémentale efficace

1.4.2.a Objectif

Dans la description du contexte scientifique du projet, nous avons motivé la raison pour laquelle l'ARR doit proposer des solutions performantes au problème de la régression relationnelle : il s'agit d'apprendre – quand les états sont décrits dans des formalismes relationnels – des approximations à la fonction valeur $V:S\rightarrow\mathbb{R}$ ou de la fonction qualité $Q:S\times A\rightarrow\mathbb{R}$. Les techniques de PLI employées pour l'approximation de fonctions devront nécessairement être incrémentales. En outre, les algorithmes d'AR utilisent souvent des versions intermédiaires de leurs approximations pour les améliorer de manière récurrente. En conséquence, les premiers apprentissages reposent souvent sur des données erronées qu'il conviendra d' « oublier » à terme. Les algorithmes d'apprentissage employés devront donc être adaptés à ce caractère instable de l'AR.

En AR, l'ordre de présentation des données aux algorithmes d'apprentissage dépend de la « trajectoire » du système dans son environnement, si bien que deux exemples successifs concernent souvent des états assez semblables. Trop bien prendre en compte le caractère instable de l'AR risque donc d'occasionner l' « oubli » de ce qui se produit dans des états plus dissemblables de l'état courant, au seul prétexte qu'ils auraient été visité plus loin dans le passé.

Le problème d'ARR pose donc de nouveaux défis à la PLI, en mettant l'accent sur l'incrémentalité et en offrant des contraintes nouvelles concernant la stabilité.

Les systèmes de l'état de l'art ont très diversement abordés ces deux problèmes. Le premier système de régression relationnelle, TILDE-RT (Blockeel & al, 1998) couplé à un système de Q-learning dans le système Q-RRL (Dzeroski & al, 2001) n'est pas incrémental : le système réapprend à la fin de chaque épisode à partir de tous les couples état/action déjà rencontrés.

Les travaux postérieurs de la KUL, notamment ceux de Driessens ont eu pour but d'améliorer le système de régression relationnelle, en mettant l'accent sur la question de l'incrémentalité. RRL-TG (Driessens et al, 2001) construit incrémentalement des arbres de régression comme TILDE. TG choisit de séparer un noeud-feuille de l'arbre en plusieurs fils :

- si ce noeud couvre suffisamment d'exemples ;
- et si un test permettant de séparer ce noeud en plusieurs fils devient hautement significatif.

En revanche, la structure d'arbre est assez rigide et s'avère peu adaptée pour faire face au problème de la dérive de concept : des noeuds peuvent être introduits dans l'arbre au début de l'apprentissage sans qu'ils ne fassent partie de l'arbre de régression optimal, et il est difficile de les remettre en cause ultérieurement.

Plus récemment, le système RIB (Driessens, 2003) adopte une approche de type apprentissage à base d'instances pour l'apprentissage de la fonction *Q*. RIB stocke un ensemble de prototypes associés à la valeur, et utilise ces prototypes pour prédire – par un algorithme de type *k-plus-proches-voisins* – de nouvelles valeurs. Il utilise donc une distance relationnelle adaptée au problème à résoudre. RIB gère de manière assez rudimentaire le problème de la dérive de concept : il élimine les prototypes qui ne lui sont pas strictement nécessaires pour réaliser une bonne prédiction, et ceux qui participent beaucoup à l'erreur de prédiction. RIB est efficace sur des problèmes simples du monde des blocs, mais nos premières expérimentations montrent qu'il est difficile de l'appliquer à des problèmes de taille plus importante. Outre la question de la distance relationnelle délicate à définir, deux autres paramètres de RIB posent problème : le nombre de prototypes et la fonction de mise à jour des prototypes stockés.

KBR (Gartner & al, 2004) repose sur la définition d'un noyau de graphe sur les couples état/action qui permet de produire des approximations très fines de la fonction valeur. En revanche, KBR ne répond pas au critère d'incrémentalité : tous les couples état/action rencontrés sont conservés, rendant l'approche très inefficace en temps de calcul.

Toutes ces approches pionnières répondent de notre point de vue de manière encore incomplète aux deux problèmes posés par la régression relationnelle dans le cadre de l'ARR : l'incrémentalité et la stabilité (dérive de concepts).

1.4.2.b Programme

Afin de concevoir un algorithme efficace et incrémental de régression relationnelle, nous nous appuierons sur les acquis de l'équipe en PLI, en particulier dans le domaine de l'apprentissage relationnel guidé par les données et utilisant des techniques de propositionnalisation (Alphonse, 2003).

Les travaux du WP2 conduiront à employer TD(λ) pour le calcul des fonctions valeur, mais en employant de nouveaux algorithmes d'apprentissage relationnel mieux adaptés au problème que RIB (Driessens & al, 2003) ou KBR (Gartner et al, 2004) qui généralise mieux mais au prix de performances moindres. Ces travaux dureront 12 mois. Ils seront validés par des publications présentant les algorithmes de PLI incrémentaux en question.

Les algorithmes développés permettront d'envisager la résolution de problèmes plus complexes que ceux abordés par TD(λ) utilisant RIB. Un travail de codage important consistera à rendre notre plateforme logicielle interopérable avec des jeux vidéo dont le code source est disponible, et de rendre disponible pour nos algorithmes des représentations relationnelles. Les jeux vidéo présentent l'intérêt de fournir des environnements dynamiques plus complexes que la plupart des environnements d'étude que nous employons ordinairement. On pourra par exemple utiliser le jeu FreeCraft comme l'avaient fait Guestrin et al (2003) pour illustrer le passage à l'échelle : une stratégie était apprise en employant quelques unités seulement, et elle s'avérait réutilisable dans des cas plus complexes.

Le produit du WP2 sera donc :

- Un algorithme efficace pour la régression relationnelle efficace, et prenant en compte les spécificités de l'AR ;
- Une validation expérimentale sur des problèmes « réels » issus du domaine des jeux vidéo.

Ces travaux seront menés par :

- Dominique Bouthinon, responsable et spécialiste d'apprentissage dans des logiques d'ordre 1 ;
- Pierre Gérard pour ses compétences en AR
- Erick Alphonse pour sa maîtrise de la plateforme logicielle ;
- Un ingénieur de recherche.

1.4.3 WP3 : AR indirect dans le cadre relationnel

1.4.3.a Objectif

Le WP2 aura conduit à la conception, l'implémentation et l'évaluation de systèmes efficaces permettant l'approximation incrémentale des fonctions de récompense immédiate, ainsi que des fonctions valeur $V : S \rightarrow \mathbb{R}$ ou de qualité $Q : S \times A \rightarrow \mathbb{R}$.

Pour aborder le problème de l'AR indirect dans le cadre relationnel, il conviendra d'être en mesure de produire également une approximation de la fonction de transition $T : S \times A \rightarrow S$. En premier lieu, nous en définirons formellement une représentation relationnelle et pour ce faire, nous exploiterons les travaux de Boutilier et al (2001) dans le cadre de la programmation dynamique. En reprenant l'exemple du monde des blocs à empiler, il faudra être en mesure de produire un modèle permettant de prédire des transitions telles que :

AuSol(a), AuSol(b), ***Empiler(b,a)*** *→ AuSol(a), Sur(b,a)*[6]

pour représenter la transition suivante :

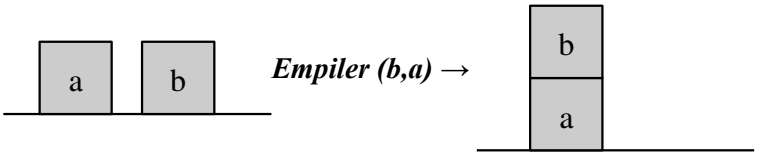


Fig 3 : Transition pour l'action Empiler(X,Y)

Le modèle de la fonction de transition devra être en mesure de produire *Sur(b,a)* et *AuSol(a)* quand on lui fournit en entrée la description de l'état de la figure 3, ***AuSol(a), AuSol(b)***, associé à l'action ***Empiler(b,a)***. L'objectif est d'apprendre un tel modèle partir d'exemples (positifs et négatifs), par exemple pour l'action *Empiler(X,Y) :*

AuSol(X), AuSol(Y), ***Empiler(Y,X)*** *→ AuSol(X), Sur(Y,X)*[7]

Dans ses travaux antérieurs portant sur l'AR indirect – mais dans un cadre propositionnel – Pierre Gérard défendait la thèse selon laquelle il était plus simple et avantageux de scinder de telles règles de prédiction en plusieurs complémentaires. Ce faisant, pour prédire la transition précédente – en utilisant la représentation propositionnelle de la section 1.2.2 : ***(0,0,1,0)(1,0)→(0,0,0,1)*** **–** on apprend des fonctions de transition partielles, de manière à prédire séparément chaque attribut :

- Une règle ***(0,0,1,0)(1,0)→ 0*** issue d'un premier module d'approximation pour l'attribut no 1 ;
- Une règle ***(0,0,1,0)(1,0)→ 0*** issue d'un second module d'approximation pour l'attribut no 2 ;
- Une règle ***(0,0,1,0)(1,0)→ 0*** issue d'un troisième module d'approximation pour l'attribut no 3 ;
- Une règle ***(0,0,1,0)(1,0)→ 1*** issue d'un quatrième module d'approximation pour l'attribut no 4 ;

Transposer cette approche dans un cadre relationnel reviendrait à chercher à prédire chaque littéral séparément, avec des règles complémentaires du type :

Empiler(Y,X) *→ Sur(Y,X).*

AuSol(X), ***Empiler(Y,X)*** *→ AuSol(X)*[8]

Ces règles peuvent chacune provenir d'un module d'approximation distinct, chacun permettant de prédire un littéral donné dans le cadre de l'application d'une action. On renverse ainsi la question « Quelle est la conséquence d'une certaine action dans un état donné ? » en la reformulant comme « Dans quels cas d'applications de *Empiler(Y,X)* produit-on *AuSol(X)* et indépendamment de cela, dans quels cas produit-on *Sur(Y,X)* ? ».

On peut ainsi chercher à apprendre dans quelles conditions chaque littéral est produit au moyen d'une fonction partielle de transition comme T_{Sur} et T_{AuSol}. La production d'une anticipation complète se fera en sollicitant les modèles de toutes les fonctions de transition partielles. Il conviendra en premier lieu d'examiner attentivement les conditions d'indépendance des différentes fonctions de transition, de manière à construire un cadre formel pour ces anticipations partielles dans le cas relationnel.

Afin d'apprendre ces fonctions partielles, les systèmes de régression incrémentale développés dans le WP2, qui permettent de prédire des scalaires et non des littéraux ne seront d'aucune utilité. Il convient de développer des systèmes d'apprentissage incrémentaux spécifiquement adaptés à ce problème.

---

[6] Si au lieu de prédire directement le nouvel état on voulait prédire les changements entre l'ancien et le nouveau – *cad* si on utilisait une fonction transition de type $T : S \times A \rightarrow \Delta S$ – on aurait la règle suivante :

AuSol(a), AuSol(a), AuSol(b), ***Empiler(b,a)*** *→ Sur(b,a),* $\neg$*AuSol(b)*

[7] Pour n'exprimer que ce qui change entre deux instants, on utiliserait la règle générale suivante :

AuSol(X), AuSol(Y), ***Empiler(Y,X)*** *→ Sur(Y,X),*$\neg$*AuSol(Y)*

[8] Ou alors :

Empiler(Y,X) *→ Sur(Y,X)*

Empiler(Y,X) *→* $\neg$*AuSol(Y)*

Des travaux récents de chercheurs européens dont Croonenborghs & al (2007) et américains issus de la communauté de planification (Pasula & al, 2004) ont récemment abordé l'apprentissage et l'exploitation d'un modèle des actions afin d'améliorer les performances (temps de convergence) de l'apprentissage par renforcement. Ce problème peut être modélisé dans le cadre de l'apprentissage supervisé où la classe à prédire est un littéral dont la valeur de vérité est modifiée par l'application de l'action (positif si l'action rend ce fait vrai ou négatif si l'action rend ce fait faux). Ce cadre est plus étudié en PLI que celui de la régression, et peut être vu comme un problème de révision incrémentale du modèle des actions (le modèle est initialement vide, et le système d'ARR, par exemple après chaque épisode, le met à jour).

Dans les travaux de Croonenborghs & al (2007), un arbre de décision est utilisé pour modéliser les conséquences des actions, mais seuls les paramètres (distribution de probabilité associée aux conséquences de chaque action) sont appris et non la structure. Pasula & al (2004) décrivent un modèle des actions assez sophistiqué, non plus sous forme d'arbre, mais sous forme de règles d'ordre 1 probabiliste et proposent un algorithme pour apprendre ces règles. Dans ce travail, sont donc apprises à la fois les règles et la distribution de probabilités associée. La robustesse de l'algorithme d'apprentissage de ces règles (très expressives) demande à être évalué à grande échelle ainsi que sa complexité.

1.4.3.b Programme

Une première partie du WP3 sera essentiellement théorique, et permettra de définir le cadre formel des travaux ultérieurs en ARR indirect, en s'appuyant sur des travaux connexes en programmation dynamique. Cette partie durera 6 mois et pourra être menée parallèlement à la fin du WP1, de manière indépendante à l'intégration de plusieurs problèmes expérimentaux, mais après avoir travaillé sur l'adaptation de TD(λ).

Le produit de cette partie du WP3 sera :

- Un rapport technique détaillant nos choix pour la représentation de la fonction de transition.

La suite du WP3 permettra d'exploiter le formalisme ainsi défini pour apprendre la fonction de transition. Ce sujet pourra être exploré indépendamment de la question de la régression incrémentale puisque l'apprentissage de la dynamique de l'environnement est indépendant des récompenses obtenues : il ne s'agit que de prédire les nouveaux états après application d'une action ; c'est un apprentissage dit « latent ». Ces travaux pourront donc débuter pendant le WP3 et dureront 12 mois. Ils seront validés par des publications présentant les algorithmes de PLI incrémentaux en question.

Ces travaux seront conduits par :

- Pierre Gérard, responsable et spécialiste d'AR ;
- Dominique Bouthinon et Aomar Osmani pour leurs compétences en apprentissage relationnel ;
- Un ingénieur de recherche.

Le WP3 permettra de produire :

- Un algorithme pour l'apprentissage d'un modèle de la fonction de transition. Les expérimentations seront menées sur plusieurs types de problèmes académiques (voir WP1) en utilisant des politiques aléatoires.

1.4.4 WP4 : Intégration de l'apprentissage de la fonction de transition et des fonctions valeur

1.4.4.a Objectif

A ce point du projet, le WP2 aura permis de perfectionner les techniques de régression incrémentale relationnelle pour produire des approximations des fonctions R, V ou Q ; et le le WP3 aura permis de proposer des algorithmes pour l'approximation de la fonction de transition T.

Pour produire un système complet d'AR indirect, il faut être en mesure d'apprendre une approximation de la fonction de transition tout en apprenant à agir au mieux, c'est-à-dire en apprenant une fonction de qualité. Au compromis exploration/exploitation classique en AR, on adjoint un dilemme entre acquisition d'un modèle de T et construction de Q ou V.

Avec un algorithme usuel en AR indirect comme DynaQ (Sutton, 1990), on utilise les modèles courants de T et de R pour produire des tuples simulés $(s, a, r'=R_t(s), s'=T_t(s))$. Ces tuples sont employés de la même manière que les tuples $(s_t, a_t, r_{t+1}, s_{t+1})$ directement issus de l'interaction avec l'environnement, de manière à améliorer la politique et les fonctions valeur. Ce faisant, on est immédiatement confronté aux deux problèmes suivants :

- au début de l'apprentissage, les modèles de T et de R sont très imparfaits et en conséquence, les exemples produits pour l'apprentissage des fonctions valeur sont très éloignés de la réalité ;
- dès que la politique s'améliore, les exemples réels (s_t, a_t, s_{t+1}) employés pour l'apprentissage de T sont principalement issus d'une séquence optimale ou presque, si bien que les données fournies en fin d'apprentissage seront sur-représentées au détriment des données concernant les conséquences des actions sous-optimales. En suivant une politique optimale, on risque de « désapprendre » T.

L'apprentissage simultané du modèle de l'environnement et des actions optimales soulève donc de nouvelles questions. Pierre Gérard (2002) avait proposé de résoudre ce double problème au moyen de l'agrégation hiérarchique de deux critères pour la qualité d'une politique : une qualité vis-à vis des récompenses, et une autre concernant la précision du modèle de l'environnement. On emploiera dans ce projet le même type d'approche consistant à employer une agrégation de critères adaptée au problème : hiérarchique, par front de Pareto etc.

Le premier algorithme employé pour l'AR indirect sera DynaQ, mais nous adapterons également des algorithmes plus sophistiqués et efficaces tels que Prioritized Sweeping (Moore and Atkeson, 1993 ; Peng and Williams, 1993).

1.4.4.b Programme

Le WP4 permettra d'intégrer les produits de WP2 et WP3. Cette intégration soulevant de nouveaux problèmes, ces travaux dureront 6 mois à compter de la fin des WP2 et WP3.

Nous validerons nos algorithmes expérimentalement sur des problèmes de complexité variable en exploitant des problèmes classiques implémentés dans le WP1, mais aussi des problèmes plus complexes issus du domaine des jeux-vidéo comme dans le WP2.

Les plus gros besoins en implémentation étant couverts après les WP2 et WP3, il ne sera plus nécessaire à ce point d'employer un ingénieur de recherche. Le WP4 sera conduit par Pierre Gérard, responsable ;

Le WP4 permettra de produire :

- Un système d'AR indirect complet de type Dyna ;
- Un système exploitant l'algorithme Prioritized Sweeping.

1.4.5 WP5 : Exploitation de la fonction de transition pour l'AR hiérarchique

1.4.5.a Objectif

Habituellement, les travaux en AR s'appuient sur des représentations « à plat » des espaces d'états et d'actions, fondées sur les Processus de Décision Markoviens (PDM). Quand la taille des espaces d'états et d'actions augmente, la complexité du PDM augmente rapidement si bien que les techniques classiques restent limitées, même lorsque l'on dispose de moyens efficaces pour effectuer les régressions incrémentales. Dans nos travaux, nous profiterons des représentations relationnelles et de leurs avantages en termes de passage à l'échelle pour aborder des problèmes de grande taille.

Une voie très étudiée en ce moment dans un cadre propositionnel est l'AR hiérarchique. Dans ce cadre, on cherche à décomposer un problème complexe « à plat » en plusieurs sous-problèmes plus simples. A un seul AR monolithique, on substitue une hiérarchie de plusieurs modules d'AR. Pour ce faire, on cherche par exemple à identifier des sous-tâches qui seront utilisées comme autant d'éléments dans des tâches de plus haut niveau. Le cadre formel des Semi-PDM permet de décomposer un PDM grâce au fait qu'une action peut ici durer plus d'un pas de temps, c'est à dire le temps qu'il faut pour exploiter un sous-PDM.

Cette approche est généralisable à tout type de représentation. Pour tirer parti de représentations relationnelles, notons que la décomposition hiérarchique d'un PDM à partir de données non structurées est un problème complexe puisqu'il nécessite d'identifier des structures dans une description propositionnelle. Dans notre approche, les relations fournissent d'emblée un premier niveau de structuration sur lequel il sera plus aisé de s'appuyer pour produire une structuration hiérarchique.

Roncagliolo & Tadepalli (2004) ont adapté MAXQ (Dietterich, 2000) au cadre relationnel. Ces travaux consistent à décomposer un MDP en regard de la fonction valeur. Or nos travaux nous auront mené à travailler sur l'apprentissage de la fonction de transition en ARR indirect. A la manière de Hengst (2002) dans le cadre propositionnel, nous chercherons donc à décomposer le PDM initial selon des critères tenant à la fonction de transition, et non sur la fonction valeur comme c'est habituellement le cas. Dans la plupart des problèmes que l'on peut étudier en AR (navigation robotique, jeux vidéo etc) la fonction de transition est souvent plus stable que les fonctions valeur. Ainsi, changer les objectifs du système en changeant les

sources de récompenses, mais dans un environnement à la dynamique semblable, le décomposition hiérarchique pourra être réutilisée. Au contraire, avec une décomposition fondée sur la fonction valeur, il serait nécessaire de tout réapprendre.

Ces derniers travaux prospectifs s'inscrivent donc dans le prolongement direct des travaux antérieurs. Ils nécessiteront de travailler sur l'abstraction temporelle comme dans Bakker et Schmidhuber (2004), si bien que nous nous appuierons sur l'état de l'art en matière de logiques temporelles et d'apprentissage de séquences.

1.4.5.b Programme

Les différentes travaux antérieurs auront permis de s'intéresser aux problèmes de l'AR relationnel et indirect. Ces deux aspects seront mis à contribution pour définir de nouvelles méthodes de factorisation de MDP pour un apprentissage hiérarchique. Ces travaux débuteront donc pendant le WP5 et se termineront après 12 mois, soit 36 mois à compter du début du projet.

Ils seront conduits par

- Pierre Gérard, responsable ;
- Aomar Osmani pour ses compétences en logiques temporelles et apprentissage de séquences.

Le WP6 permettra de produire :

- Une nouvelle méthode de factorisation des MDP exploitant la structure des représentations relationnelles, ainsi que la fonction de transition.

1.5 Résultats escomptés et retombées attendues

L'objectif du projet HARRI est donc de produire des algorithmes pour l'AR relationnel, indirect et hiérarchique. L'AR relationnel est un domaine émergeant qu'il convient d'investir au plus vite. Au niveau international, les équipes actives dans ce domaine sont spécialistes de PLI. Le LIPN – et l'équipe A^3 – a la particularité de regrouper des membres issus des communautés PLI et AR, souvent distinctes. Cette spécificité permettra de s'appuyer sur des compétences plus variées qu'à l'ordinaire pour aborder rapidement et dans de bonnes conditions un domaine encore peu étudié. Le premier objet de ce projet est donc académique, avec la production rapide de quelques travaux de référence, de manière à accroître encore la visibilité internationale du LIPN.

Comme indiqué dans la description des différents Work Packages, l'effort de développement sera également conséquent pour :

- étendre la plateforme d'apprentissage relationnel du LIPN ;
- intégrer tous les problèmes usuels en ARR ;
- interfacer ces développements avec des jeux vidéo dont le code source est disponible

Outre les publications, nous aurons produit des conditions optimales pour développer une activité à long terme, avec la possibilité de mener rapidement des expérimentations au moyen de la plateforme d'apprentissage relationnel du LIPN.

1.6 Organisation du projet

Tâche	Année 1 Year 1		Année 2 Year 2		Année 3 Year 3	
	6	12	18	24	30	36
WP1 : Adaptation d'algorithmes classiques en AR au cas relationnel *Responsable :* Pierre Gérard *Impliqués :* Erick Alphonse, Dominique Bouthinon						
WP2 : PLI pour une régression incrémentale efficace *Responsable :* Dominique Bouthinon *Impliqués :* Pierre Gérard, Erick Alphonse						
WP3 : AR indirect dans le cadre relationnel *Responsable :* Pierre Gérard *Impliqués :* Dominique Bouthinon, Aomar Osmani						
WP4 : Intégration de l'apprentissage de la fonction de transition et des fonctions valeur *Responsable :* Pierre Gérard						
WP5 : Exploitation de la fonction *T* pour l'AR hiérarchique *Responsable :* Pierre Gérard *Impliqués :* Aomar Osmani						
Rapports d'avancement semestriel	✍	✍	✍	✍	✍	✍
Rapport final						☺

✍ : Rapport d'avancement semestriel/6 month-progress report

☺ : Rapport de synthèse et récapitulatif des dépenses/Final report and expenses summary

Le WP5 est le plus prospectif et présente les risques les plus importants. Le risque est en revanche bien maîtrisé pour les autres Work Packages.

TABLEAU des LIVRABLES et des JALONS		
Tâche	Intitulé et nature des livrables et des jalons	Date de fourniture nombre de mois à compter de T0
WP1 : Adaptation d'algorithmes classiques en AR au cas relationnel		
	Plateforme logicielle pour l'AR, intégration à PROPAL	4
	Publication concernant TD(λ) relationnel sur le monde des blocs	8
	Plateforme expérimentale intégrant la plupart des problèmes utilisés en ARR	12
	Publication sur l'application de TD(λ) à des problèmes plus complexes	12
WP2 : PLI pour une régression incrémentale efficace		
	Rapport détaillant nos choix pour la représentation de la fonction de transition.	12
	Publication sur l'apprentissage d'un modèle de la fonction de transition (politique aléatoire) pour le monde des blocs	18
	Publication sur l'apprentissage de T dans des environnements plus complexes	24
WP3 : AR indirect dans le cadre relationnel		
	Publication concernant TD(λ) relationnel avec un nouvel algorithme de régression sur le monde des blocs	16
	Publication concernant TD(λ) relationnel avec un nouvel algorithme de régression sur d'autres problèmes	20
	Plateforme d'expérimentation interopérable avec FreeCraft (jeu vidéo)	20
	Publication pour l'expérimentation de TD(λ) pour le problème de Freecraft : passage à l'échelle	24
WP4 : Intégration de l'apprentissage de la fonction de transition et des fonctions valeur		
	Publication de l'intégration de WP2 et WP3 : système complet d'ARRI	28
	Publication concernant le passage à l'échelle de notre nouveau système	30
	Mise à disposition de tous les codes source	30
WP5 : Exploitation de la fonction de transition pour l'AR hiérarchique		
	Rapport arrêtant nos choix théoriques	34
	Publication des premiers résultats dans le monde des blocs pour HARRI	36

1.7 Organisation

1.7.1 Constitution de l'équipe proposée

Le Laboratoire d'Informatique de Paris Nord (LIPN) est une UMR CNRS (#7030) associée à l'Université de Paris13. Il compte 4 équipes dont une composée exclusivement de spécialistes d'apprentissage artificiel : l'équipe A^3 (Apprentissage Artificiel et Applications). Cette équipe a été créée en 2003, après un recentrage des activités de l'ancienne équipe ADAge (Apprentissage, Diagnostic et Agents) autour de la seule activité « Apprentissage ». Elle compte 12 membres permanents, ce qui en fait une des plus grandes équipes françaises d'apprentissage. Ses activités rentrent dans le cadre du pôle de compétitivité « Cap Digital ».

L'équipe A^3 dispose de deux axes de recherche principaux : l'apprentissage symbolique et l'apprentissage numérique. De manière à développer les synergies entre les composantes de l'équipe, le LIPN souhaite – comme cela a été annoncé lors de la dernière évaluation CNRS du laboratoire en janvier 2008 – développer un axe transversal. Le développement d'une activité AR – habituellement associé à l'apprentissage numérique – mais employant des représentations issues d'une tradition symbolique – s'intègre donc parfaitement à la politique du laboratoire.

Le projet proposé ici a donc vocation à être pérennisé au sein de l'équipe et du laboratoire puisqu'il marque la première étape du développement de cet axe de recherche transversal.

1.7.2 Complémentarité et synergie des membres de l'équipe

Pierre Gérard, jeune maître de conférences et porteur du projet, est entré dans l'équipe en 2003. Il est spécialiste d'apprentissage par renforcement. Le projet HARRI s'inscrit dans le prolongement naturel de ses travaux de thèse, mais dans un cadre relationnel au moyen de la PLI.

Erick Alphonse a été recruté comme MCF en 2005. Il a conçu la plateforme d'apprentissage relationnel du LIPN autour de son système PROPAL, cette plateforme est destinée à fédérer les différents développements de l'équipe en apprentissage relationnel, de manière à favoriser les synergies. Le développement de l'axe transversal autour de l'apprentissage par renforcement relationnel permettra donc d'intégrer les plus récents recrutements autour d'un projet fédérateur commun.

Dominique Bouthinon (MCF depuis 1999) et Aomar Osmani (MCF depuis 2000) sont les jeunes chercheurs les plus expérimentés de l'équipe du projet HARRI. Ils sont spécialistes d'apprentissage symbolique et d'apprentissage relationnel en particulier :

- Dominique Bouthinon travaille sur l'apprentissage dans des logiques d'ordre 1 ;
- Aomar Osmani travaille sur la PLI et également sur l'apprentissage de séquences et l'abstraction de séquences.

L'équipe A^3 est une des seules en Europe qui soit en mesure de développer simultanément les principaux axes de recherche ouverts par l'ARR puisqu'elle dispose à la fois de spécialistes en AR et de spécialistes en PLI. Elle est la seule en France ; son positionnement est donc à la fois singulier et stratégique pour investir un domaine de recherche émergent.

1.7.3 Qualification du responsable scientifique et des membres de l'équipe *:* résumé et CV

Le tableau ci-dessous récapitule les implications des différents participants au projet HARRI.

Prénom, NOM	Statut	Implication	Mois.Hommes (éq TP)
Pierre GERARD	**MCF**	**90% sur 36 mois**	**32**
Dominique BOUTHINON	MCF	50% sur 36 mois	18
Aomar OSMANI	MCF	25% sur 36 mois	9
Erick ALPHONSE	MCF	25% sur 36 mois	9
		Total	**68 (1,9 éq TP)**

1.7.3.a Pierre GERARD

État civil : Homme, 34 ans

Implication dans le projet : 90%

Situation actuelle : Maître de Conférences à l'IUT de Villetaneuse (UP13) depuis 2003

Autres expériences professionnelles :

- Doctorat en convention CIFRE, avec Dassault Aviation (St Cloud)

Domaines de recherche :

- Apprentissage par renforcement (direct/indirect)
- Systèmes de classeurs

Cursus universitaire :

- Doctorat en Informatique de l'Université de Paris 6
- DEA d'informatique de l'Université de Nancy I

Séjours à l'étranger :

- Séjour post-doctoral court (2 mois) à l'AIST, Tokyo, Japon

Publications récentes et liées au projet HARRI :

- Gérard, P., Meyer J.-A. and Sigaud, O. (2005) *Combining Latent Learning and Dynamic Programming in MACS*, European Journal of Operational Research 160 :614-637.
- Gérard, P., Stolzmann, W. and Sigaud, O. (2002) *YACS : a new Learning Classifier System using Anticipation* Journal of Soft Computing : Special Issue on Learning Classifier Systems. 6 (3-4), 216-228 Springer Verlag.
- Butz, M.V., Sigaud, O. and Gérard, P. (Eds) (2003) *LNCS 2684 : Anticipatory Behavior in Adaptive Learning Systems*
- Gérard, P. and Sigaud, O. (2003) *Designing Efficient Exploration with MACS : Modules and Function Approximation*, Proceedings of the Genetic and Evolutionary Computation Conference,GECCO'03.
- Gérard, P. and Sigaud, O. (2004) *Apprentissage par renforcement indirect dans les systèmes de classeurs.* JEDAI.

Quantité de publications :

- Revues internationales : 2
- Congrès internationaux avec comité de sélection : 8 (+3 LNAI)

Responsabilités administratives :

- Membre élu du Conseil d'Administration de l'Université
- Membre de la Commission de Spécialistes – section 27

1.7.3.b Dominique Bouthinon

État civil : Homme, 49 ans

Implication dans le projet : 50%

Situation actuelle : Maître de Conférences à l'IUT de Villetaneuse (UP13) depuis 1999

Autres expériences professionnelles :

- Ingénieur CNRS, 1988-1999 (en informatique)

Domaines de recherche :

- Apprentissage symbolique
- Programmation Logique Inductive
- Apprentissage relationnel fondé sur la propositionnalisation

Cursus universitaire :

- Doctorat de l'Université de Paris 13
- Ingénieur CNAM en informatique

Publications liées au projet HARRI :

- Krief, F. et Bouthinon, D. (2004) *A Learning and Intentional Local Policy Decision Point for Dynamic QoS Provisioning*, IFIP TC6, Third International Conference on Network Control and Engineering for QoS, Security and Mobility, NetCon 2004, pp. 277-288, Springer edition, November 2-5, 2004, Palma de Mallorca, Spain
- Dominique Bouthinon (2002) *Mesure de l'information apportée par différents types d'exemples et de biais inductifs* Conférence francophone d'Apprentissage, Orléans, 17-19 juin 2002.
- Bouthinon, D. and Soldano, H. (1998) *An Inductive Logic Programming Framework to Learn a Concept from Ambiguous Examples.* ECML 1998 : 238-249 . Proceedings. Lecture Notes in Computer Science 1398 Springer 1998, ISBN 3-540-64417-2

1.7.3.c Aomar Osmani

État civil : Homme, 36 ans

Implication dans le projet : 25%

Situation actuelle : Maître de Conférences à l'IUT de Villetaneuse (UP13) depuis 2000

Domaines de recherche :

- Apprentissage relationnel
- Apprentissage de séquences

Cursus universitaire :

- Doctorat en Informatique de l'Université de Paris 13
- DEA IARFA (Intelligence Artificielle, Reconnaissance des Formes et Applications) de l'Université de Paris 6, institut Blaise Pascal.
- Diplôme d'Ingénieur d'état en Informatique à l'Institut National d'informatique de Tiziouzou

Publications récentes et liées au projet HARRI :

- Alphonse, E. and Osmani, A. (2008) *On the connection between the phase transition of the covering test and the learning success rate in ILP*. Machine Learning Journal (MLJ) - 2008, Édition spéciale à paraître
- Alphonse, E. and Osmani, A. (2006) *Using near-misses to cross plateaus : a study about phase transition in relational learning.* The 16th International Conference on Inductive Logic Programming (ILP 2006).
- Bouandas, K. and Osmani, A. (2008) *Mining Temporal Sequences using Interval Algebra* INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY AND INTELLIGENT COMPUTING (to appear, april)
- Bouandas, K. and Osmani, A. (2007) K. Bouandas and A. Osmani Mining *Association Rules in Temporal Sequences in Computational Intelligence and DataMining* CIDM'2007 pages 610-615, published by IEEE press. ISBN : 1-57735-177-0
- Asservatham, S., Osmani, A. ans Viennet, E. (2006) *bitSPADE : A Lattice-Based Sequential Pattern Mining Algorithm Using Bitmap Representation*, 2006 IEEE International conference on Data Mining (ICDM'06)

1.7.3.d Erick Alphonse

État civil : Homme, 34 ans

Implication dans le projet : 25%

Situation actuelle : Maître de Conférences à l'Université de Paris 13 depuis 2005

Domaines de recherche :

- Apprentissage relationnel

Cursus universitaire :

- Doctorat de l'Université Paris 11
- DEA de Contrôle des Systèmes à l'UTC
- Diplôme d'Ingénieur en Informatique de l'INSA de Rennes, spécialité Systèmes et Réseaux

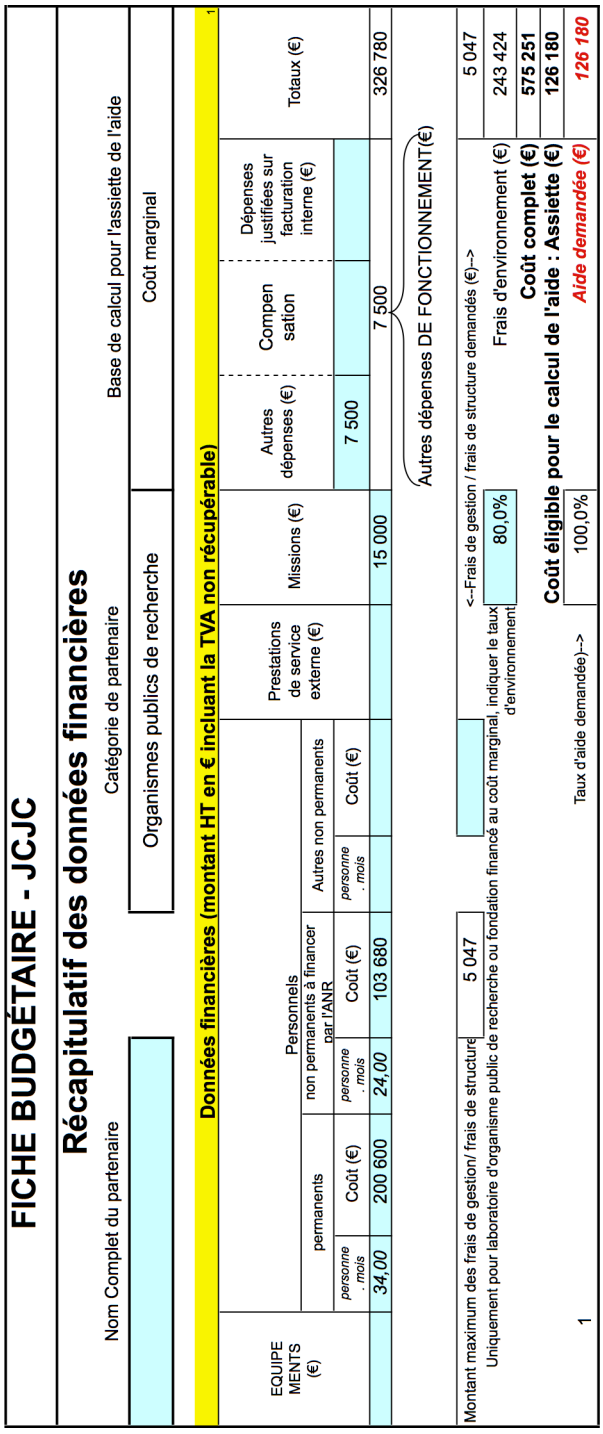
Publications récentes et liées au projet HARRI :

- Alphonse, E. and Osmani, A. (2008) *On the connection between the phase transition of the covering test and the learning success rate in ILP*. Machine Learning Journal (MLJ) - 2008, Édition spéciale à paraître
- Alphonse, E. and Osmani, A. (2007) *Phase Transition and Heuristic Search in Relational Learning*. ICMLA 2007, À paraître
- Alphonse, E. and Rouveirol, C. (2006) *Extension of the Top-Down Data-Driven Strategy to ILP*. Int. Conference on 16th International Conference on Inductive Logic Programming (ILP 2006), LNAI 4455
- Alphonse, E. and Matwin, S. (2004) *Filtering Multi-instance Problems to Reduce Dimensionality in Relational Learning*, Int. Journal of Intelligent Information Systems
- Alphonse, E. (2004) *Macro-operators revisited in ILP,* Int. Conf. on Inductive Logic Programming (ILP'04), LNCS 3194

1.7.4 Accès aux grands instruments

Aucun grand instrument n'est nécessaire pour le projet HARRI.

1.7.5 Fiche budgétaire



FICHE BUDGÉTAIRE - JCJC

Récapitulatif des données financières

Nom Complet du partenaire

Catégorie de partenaire: Organismes publics de recherche

Base de calcul pour l'assiette de l'aide: Coût marginal

Données financières (montant HT en € incluant la TVA non récupérable) 1

EQUIPEMENTS (€)	Personnels permanents personne . mois	Personnels permanents Coût (€)	Personnels non permanents à financer par l'ANR personne . mois	Personnels non permanents à financer par l'ANR Coût (€)	Autres non permanents personne . mois	Autres non permanents Coût (€)	Prestations de service externe (€)	Missions (€)	Autres dépenses (€)	Compensation	Dépenses justifiées sur facturation interne (€)	Totaux (€)
	34,00	200 600	24,00	103 680				15 000	7 500			326 780

Autres dépenses DE FONCTIONNEMENT(€) 7 500

Montant maximum des frais de gestion/ frais de structure 5 047

<--Frais de gestion / frais de structure demandés (€)--> 5 047

Uniquement pour laboratoire d'organisme public de recherche ou fondation financé au coût marginal, indiquer le taux d'environnement 80,0%

Frais d'environnement (€) 243 424

Coût complet (€) 575 251

Coût éligible pour le calcul de l'aide : Assiette (€) 126 180

1 Taux d'aide demandée)--> 100,0%

Aide demandée (€) 126 180

2 Justification scientifique des moyens demandés

2.1 Équipement

Aucun équipement d'un montant supérieur à 4000 euros n'est nécessaire pour le projet HARRI.

2.2 Personnel

Le projet HARRI vise à positionner un laboratoire français dans un domaine de recherche émergent de manière à produire rapidement des travaux de référence. Il sera nécessaire de mener rapidement de multiples expérimentations pour valider ou invalider des hypothèses.

Pour être suffisamment réactif, il convient de disposer d'une plateforme logicielle souple et prenant en charge de manière optimale la présentation et l'exploitation des résultats expérimentaux. L'équipe A^3 a engagé cet effort avec la plateforme logicielle développée autour de PROPAL et nous continuerons à fédérer les développements de l'équipe autour de ce système, de manière à favoriser les synergies entre les différents projets de l'équipe.

Le projet HARRI s'inscrit dans cette politique et pour le mener à bien, il convient de mobiliser des ingénieurs autant que des chercheurs. C'est pourquoi le projet HARRI nécessite le financement d'un personnel extérieur chargé des développements les plus lourds. Ces développements devant être pour leur plus grande part achevés après 24 mois, nous demandons le financement d'un ingénieur de recherche pour cette durée.

Les grandes lignes du profil de cet ingénieur seront :

- Ingénieur de recherche 2ème Classe
- Familiarité avec le domaine de l'apprentissage automatique
- Bonne connaissances techniques pour l'interopérabilité de systèmes hétérogènes (SWI Prolog, C++ etc)
- Familiarité avec les jeux vidéo

2.3 Prestation de service externe

Aucun service externe n'est nécessaire pour le projet HARRI.

2.4 Missions

Le projet HARRI vise à produire rapidement des travaux de référence en apprentissage par renforcement relationnel. Outres les journaux du domaine de l'apprentissage, nous aurons donc une politique de publication active dans les conférences et workshops internationaux, de manière à accélérer la visibilité de nos travaux. Nous pourrons valoriser nos travaux scientifiques à la fois dans les conférences habituelles pour l'ILP et celles familières des spécialistes de l'apprentissage par renforcement. Nous envisageons donc des déplacements internationaux fréquents pour des conférences et workshops internationaux.

En outre, nous nous appuierons sur le projet HARRI pour développer nos collaborations avec l'Université Catholique de Leuven, référence internationale incontournable en ce qui concerne la programmation logique inductive. Cette collaboration nécessitera de séjourner parfois en Belgique.

Pour le projet HARRI, nous demandons donc des frais de mission supérieurs aux 5% octroyés sans justification. Ces frais seront nécessaires pour financer des déplacements pour des colloques, ou des collaborations telles que celle que nous entendons développer avec l'Université Catholique de Leuven.

2.5 Dépenses justifiées sur une procédure de facturation interne

Aucune dépense justifiée sur une procédure interne n'est nécessaire pour le projet HARRI.

2.6 Autres dépenses de fonctionnement

Les autres dépenses de fonctionnement ne concernent que l'acquisition de petits matériels comme un ordinateur portable, par exemple.

Annexes

Description des participants

(cf. 1.7.1)

Biographies

(cf. 1.7.3)

Implication des personnes dans d'autres contrats

Nom de la personne participant au projet	Personne. Mois	Intitulé de l'appel à projets Source de financement Montant attribué	Titre du projet	Nom* du coordinateur	Date début -Date fin

Demandes de contrats en cours d'évaluation

Nom de la personne participant au projet	Personne. Mois	Intitulé de l'appel à projets Source de financement Montant demandé	Titre du projet	Nom* du coordinateur