



HAL
open science

STAR: Un outil permettant une intégration efficace de modèles cognitifs simples en EIAH

Bruno Martin, Emmanuel Sander, Jean-Marc Labat, Jean-François Richard

► **To cite this version:**

Bruno Martin, Emmanuel Sander, Jean-Marc Labat, Jean-François Richard. STAR: Un outil permettant une intégration efficace de modèles cognitifs simples en EIAH. Environnements Informatiques pour l'Apprentissage Humain, May 2013, Toulouse, France. pp.293-304. hal-00949170

HAL Id: hal-00949170

<https://hal.science/hal-00949170v1>

Submitted on 19 Feb 2014

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

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

STAR : Un outil permettant une intégration efficace de modèles cognitifs simples en EIAH

Martin Bruno*, Sander Emmanuel*, Labat Jean-Marc**, Richard Jean-François*

* Laboratoire Paragraphe, Université Paris-8

2 Rue de la Liberté

93526 Saint-Denis Cedex 02

martin.bruno.mail@gmail.com

sander@univ-paris8.fr

richard@univ-paris8.fr

** LIP6, Université Pierre et Marie Curie

4 Place Jussieu

75005 Paris

jean-marc.labat@lip6.fr

RÉSUMÉ. Notre article présente STAR acronyme de *Simple Toolbox to Analyze Reasoning*, un cadre pour écrire, utiliser et évaluer la pertinence un modèle cognitif déterministe minimaliste en EIAH. Par STAR nous amorçons une réponse à deux problématiques présentes en modélisation cognitive en EIAH : réduire le temps de mise en place d'un modèle de l'élève, et mesurer la fiabilité statistique d'un modèle en prenant en compte le nombre de degrés de liberté de ce dernier. STAR est un outil générique adapté au modèle des contraintes, pour faciliter son intégration dans les environnements d'apprentissage. Nous présentons dans cet article les solutions informatiques et statistiques portées par STAR.

MOTS-CLÉS : model selection, modélisation cognitive, EIAH, arithmétique

1. Modèles cognitifs computationnels

1.1. Généralités

1.1.1. Qu'est-ce qu'un modèle cognitif computationnel ?

Les sciences cognitives sont un ensemble de disciplines qui ont pour objet la description et l'explication de la pensée humaine, particulièrement dans des tâches complexes mettant notamment en jeu la perception, l'apprentissage ou la résolution de problème. La complexité de ces objets d'études met les modèles cognitifs au premier plan de ces recherches car ils permettent de poser un cadre explicatif et descriptif pour comprendre et donner sens aux observations. Les modèles cognitifs computationnels sont souvent distingués des modèles cognitifs conceptuels du fait de leurs niveaux de description respectifs. Les modèles conceptuels sont la description en langage naturel d'un point de vue théorique tandis que les modèles computationnels font un pas de plus en formalisant mathématiquement et algorithmiquement le processus qu'ils cherchent à décrire. L'avantage de ces derniers est qu'ils permettent de reproduire le phénomène à modéliser ce qui permet de mesurer leur pouvoir explicatif et de faire des prédictions quantitatives. Le développement de l'informatique a facilité leur utilisation, permettant d'écrire et d'exécuter les programmes informatiques conçus à cette fin.

1.1.2. Quel est l'intérêt des modèles cognitifs computationnels en EIAH ?

La plupart des modèles computationnels ont des paramètres inconnus qui permettent dans le cadre d'un EIAH de distinguer les apprenants entre eux. Leurs valeurs sont généralement déterminées par une minimisation de la distance entre les productions du modèle et les productions réelles de l'apprenant. Les paramètres rendent compte des différences interindividuelles mais sont aussi des degrés de liberté dont il convient de minimiser le nombre sans quoi le modèle perdrait de son pouvoir explicatif. Cette problématique est développée plus loin dans ce texte.

1.2. Le modèle des contraintes

1.2.1. Une alternative aux règles de production

Le modèle des contraintes [RICHARD et al. 93] a pour racine la formulation cognitiviste d'un cadre d'analyse pour la résolution de problème. Développée par Newell et Simon dans les années 1970 [SIMON & NEWELL 71], cette formalisation consiste à représenter la résolution comme le déplacement dans un graphe appelé espace de recherche. Les nœuds sont les états du problème, les stratégies en sont les règles de parcours. Cette conception a été déterminante et a permis l'écriture de modèles cognitifs computationnels portant sur la résolution de problème. ACT-R [ANDERSON 93] est l'héritier le plus connu de cette approche et a largement investi le champ applicatif de l'apprentissage avec les Intelligent Tutoring System (ITS). Si l'approche des règles de production (règles du type [*si* <Etat> alors <Action>]) a permis de modéliser avec succès les connaissances procédurales, le bilan est plus partagé au sujet de la modélisation des connaissances déclaratives, des représentations et des interprétations du problème [OHLSSON 1990]. Le modèle des contraintes emploie un formalisme contrastant avec celui des systèmes de règles de production car il est constitué de règles du type [*si* <Etat> alors **Non** <Action>] appelées contraintes. Cette différenciation provient d'une conception différente de la résolution de

problème selon laquelle le sujet est soumis à un ensemble de contraintes de nature diverse qui restreignent, voire déterminent son processus de résolution. L'accent est non plus mis sur les compétences procédurales, mais sur les représentations que le sujet a du problème. Les contraintes sont réparties en familles, une seule contrainte (ou aucune) peut être sélectionnée par famille. Cela revient donc à un **format de description de type attributs-valeurs** dont l'attribut correspond à la famille et la valeur est portée par la contrainte sélectionnée.

1.2.2. Exemple avec l'arithmétique en classe élémentaire.

STAR est né dans un contexte de recherche consistant à coupler l'EIAH DIANE [HAKEM 05] qui porte sur l'arithmétique en classe élémentaire, au modèle des contraintes pour la détermination des profils cognitifs des apprenants. C'est la raison pour laquelle nous nous appuyons sur le domaine de l'arithmétique pour décrire le fonctionnement du modèle des contraintes. Le but est de modéliser la résolution de problèmes à énoncés verbaux du type « *Jean a 8 billes. Carine a 5 billes de plus que Jean. Combien Carine a-t-elle de billes ?* » Ce type de problème est assez différent des problèmes de calcul du type « $8+5= ?$ ». Les problèmes à énoncés verbaux décrivent une situation dont les relations mathématiques entre les quantités doivent être comprises pour la résolution et traduites par un (ou plusieurs) calcul(s). Ces problèmes ont un fort intérêt pédagogique car ils connectent les opérations à leurs sens pratique, mais introduisent des difficultés supplémentaires (linguistiques et logico-mathématiques [CUMMINS 91]) et peuvent induire des stratégies de résolution alternatives ou des heuristiques pas toujours pertinentes provoquées par le stéréotypage des énoncés (utiliser tous les nombres, se baser sur des mots-clefs, faire des additions par défaut...). Modéliser cette activité requiert un cadre suffisamment ouvert pour que l'on puisse exprimer et identifier dans un même temps les compétences du sujet par rapport au domaine du problème et donc ses déficits, mais aussi d'autres sources d'erreurs que sont des interprétations erronées ou l'utilisation de connaissances pragmatiques inappropriées. C'est ce que nous entreprenons avec le modèle des contraintes. Notre modélisation porte donc sur différentes composantes de la résolution de problèmes arithmétiques, les contraintes ont alors été conçues pour couvrir ces différents aspects. Pour simplifier l'exemplification, nous présentons ici une sélection de trois familles de contraintes.

La première famille, portant sur le niveau logico-mathématique de l'élève est définie par l'échelle de difficulté existant entre les problèmes à une étape du fait de la place de la variable inconnue dans la relation mathématique décrite dans l'énoncé. En reprenant la question ci-dessus « *Jean a 8 billes. Carine a 5 billes de plus que Jean. Combien Carine a-t-elle de billes ?* », il est possible de créer d'autres versions conservant la relation de comparaison mais augmentant le niveau de difficulté du problème : On peut demander le nombre de billes d'écart entre Jean et Carine connaissant le nombre de billes qu'ils ont chacun en possession ou le nombre de billes de Carine en indiquant l'écart par « *Jean a 5 billes de moins que Carine* », c'est la version la plus difficile. Pour une même relation il existe donc 3 niveaux de difficulté connus dans la littérature depuis [KINTSCH 85]. Une première famille est donc posée pour décrire la capacité que l'élève a de conceptualiser le type de relation en jeu dans le problème.

Une deuxième porte sur l'interprétation de la relation de comparaison (« *plus de* ») qui peut se retrouver régulièrement erronée et être comprise en valeur absolue, ce qui se traduirait en reprenant l'exemple par : « *Carine a 5 billes* ».

La troisième porte sur la justification ou la formulation des résultats que l'élève apporte à son calcul.

Famille 1 : compétence logico-mathématique	Famille 2 : interprétation de « d de plus »	Famille 3 : formulation des résultats
<u>Contrainte 1a</u> : capacité de conceptualiser les 3 versions de ce problème	<u>Contrainte 2a</u> : « d de plus » interprété en valeur absolue	<u>Contrainte 3a</u> : formulation systématique.
<u>Contrainte 1b</u> : capacité de conceptualiser les 2 versions les plus facile de ce problème	<u>Contrainte 2b</u> : « d de plus » interprété correctement	<u>Contrainte 3b</u> : jamais aucune formulation.
<u>Contrainte 1c</u> : capacité de conceptualiser la version la plus facile		<u>Contrainte 3c</u> : formuler uniquement quand la quantité calculée est celle de la question.
<u>Contrainte 1d</u> : incapacité de résoudre l'ensemble des versions de ce problème		

Tableau 1. Familles des contraintes sélectionnées pour l'exemple

À partir d'une analyse approfondie des erreurs observées, nous avons listé un ensemble de réponses attendues pour le problème « Jean a 8 billes. Carine a 5 billes de plus que Jean. Combien Carine a-t-elle de billes ? ».

1	8+5=13 Carine a 13 billes
2	Carine a 3 billes
3	Carine a 5 billes
4	8-5=3
5	8+5=13
6	5
7	8
8	(pas de réponse)

Tableau 2. Réponses possibles du problème donné en exemple

Cette liste amène à un certain nombre de remarques :

(1) Tous les types de solutions ne sont pas représentés ici, des sous-variantes sont possibles en prenant en compte la qualité des calculs ou le choix de l'opération (prévoyant un calcul mental, ou le choix de poser les soustractions par des additions à trou). D'autres familles de contraintes, non représentées dans cet exemple, permettent de rendre compte de ces aspects.

Le contexte est le suivant : un élève résout un certain nombre de problèmes ce qui permet de lui associer un **profil cognitif** (c'est-à-dire l'ensemble de contraintes qui simule au mieux ses protocoles de résolution). Pour chaque famille une contrainte au plus est sélectionnée. Supposons maintenant qu'un élève comporte dans son profil les contraintes 1c et 2a. La quantité « 5 » est attendue en résultat mais la formulation de sa réponse reste indéterminée car aucune contrainte n'a été choisie dans la troisième famille. Le modèle fait donc la prédiction que cet élève va fournir la réponse 3 ou 6.

Comme illustré sur cet exemple, les contraintes des différentes familles s'agrègent pour dégager parmi les **réponses possibles** un ensemble plus petit de réponses dites **réponses prédites**. Les prédictions du modèle peuvent donc être multiples. Cette terminologie est conservée dans la partie qui suit. Même si les contraintes portent sur des aspects différents

de la résolution de problèmes, il est important de noter qu'elles n'ont pas à être totalement indépendantes, par exemple la contrainte 2a et la contrainte 1a ne peuvent être vraies ensemble sur les problèmes présentés.

2. Problématique

2.1. Le problème des degrés de liberté.

Pour résumer les propos précédents, un modèle cognitif ne peut proposer de diagnostic s'il ne dispose pas au préalable de degrés de liberté (les paramètres du modèle) dont l'instanciation permet d'établir le profil cognitif de l'élève. Dans le cadre du modèle des contraintes les degrés de liberté se situent dans le choix des contraintes. Comme vu précédemment, celles-ci sont organisées en familles de contraintes exclusives les unes des autres. Le set de contraintes qui modélise l'élève est obtenu par la sélection d'au plus une contrainte par famille et est appelé le profil cognitif de l'élève.

L'existence de degrés de liberté induit un double questionnement. Le premier est celui du risque d'overfitting, traduit parfois en français par « sur-ajustement » qui traduit l'idée qu'un modèle avec beaucoup de degrés de liberté a plus de chances d'être précis même s'il est à peine meilleur que le hasard. Le cas extrême : un modèle avec autant de degrés de liberté que d'observations aurait alors une précision parfaite, mais un pouvoir prédictif nul. Dans le cadre du modèle des contraintes, il est important de déterminer si les contraintes ont un réel pouvoir explicatif, d'où la nécessité d'un critère prenant à la fois en compte le nombre de degrés de liberté du modèle et sa précision face aux données.

Le deuxième problème est un problème d'instanciation des degrés de liberté. On désire choisir le profil correspondant « le mieux » aux protocoles de résolution de l'élève, mais cette notion peut être ambiguë dans le cas des modèles déterministes qui réalisent de multiples prédictions. Il faut, en effet, à la fois minimiser le nombre de prédictions alternatives fournies par le modèle et le nombre de cas où la réponse de l'élève est hors des prédictions. Ces deux critères jouent de manière inverse : un modèle faisant des prédictions uniques a plus de chances de commettre des erreurs qu'un modèle faisant des prédictions multiples.

2.2. Le problème du temps de développement

Notre objectif¹ initial était de développer une version générique du modèle des contraintes, c'est-à-dire une bibliothèque de fonctions liée au modèle des contraintes permettant de gagner en visibilité et en temps de développement lors du déploiement sur de nouveaux domaines d'études et dans de nouveaux environnements notamment les EIAH comme DIANE. Ce projet d'envergure commence avec une version minimaliste du modèle des contraintes sans envisager immédiatement les contraintes dynamiques (qui évoluent au cours de la résolution et permettent de générer les protocoles de résolutions pas à pas). La problématique de réduction du temps de développement d'un modèle cognitif fait sens dans le contexte actuel de la recherche en EIAH puisque sa mise en place représente un coût temporel important dont la rentabilité est parfois remise en cause [ANDERSON 95]. La

¹ Thèse en cours, financée par la Direction Générale de l'Armement dont l'intitulé est « Conception et implémentation d'un modèle générique basé sur le modèle des contraintes ».

récente montée en puissance des Constraint-Based Tutor² [MITROVIC 01] témoigne en effet qu'il n'est pas nécessaire de recourir à une modélisation cognitive profonde de l'élève pour construire un tuteur efficace. Nous avons donc développé une tierce application visant une écriture et une évaluation non excessivement coûteuse d'un système de règles applicable au modèle des contraintes, sans toutefois s'y restreindre.

3. Présentation de STAR

3.1. Un critère statistique proposé

3.1.1. Méthodes classiques

Certains critères existent pour concilier précision et degrés de liberté d'un modèle, dont l'étude sont l'objet du champ de recherche nommé « model selection ». Les plus utilisés sont l'Akaike Information Criterion (AIC), le Bayesian Information Criterion (BIC) et le Minimum Description Length (MDL). À notre connaissance ils sont tous construits pour s'appliquer à des modèles probabilistes et ne s'appliquent à des modèles déterministes que par le biais d'une fonction d'erreur, celle-ci permettant de construire une version probabiliste du modèle en assignant une probabilité fictive à chaque événement.

Cependant l'évaluation d'un modèle déterministe sous une version probabiliste rend caduque le choix initial de construire un modèle déterministe, en effet on peut se poser la question de l'utilité de la construction d'un modèle déterministe si on ne peut évaluer que sa version probabiliste. De plus la construction d'une fonction d'erreur, étape cruciale, peut amener à une part d'arbitraire importante et n'assure pas l'utilisation pratique des critères précédents dans la mesure puisque des calculs assez lourds peuvent être requis pour permettre cette approche (voir [GRUNWALD 99] dans le cas du MDL).

3.1.2. Notre proposition

Notre proposition consiste à reprendre l'idée de complexité stochastique telle qu'elle est décrite chez [RISSANEN 86], mais en l'adaptant à la classe des modèles déterministes.

"[...] the stochastic complexity of a string of data, relative to a class of probabilistic models, is defined to be the fewest number of binary digits with which the data can be encoded by taking advantage of the selected models."

Mesurer la qualité d'un modèle par un calcul de complexités stochastiques est à la base du Minimum Description Length [MYUNG & PITT 05]. Cependant, comme indiqué plus haut, ces critères ne concernent que les modèles probabilistes et ne s'appliquent aux modèles déterministes que par le recours à une fonction d'erreur.

L'idée est de reprendre cette définition mais en évitant l'approche probabiliste, et en se basant sur l'idée qu'un modèle déterministe permet de décrire une version résumée des données dont il doit rendre compte. Supposons pour commencer que le modèle ne fasse pas de fausses prédictions, c'est-à-dire que les réponses prédites proposées soient toujours en accord avec les observations. Dans ce cas un moyen simple d'encoder les données par le biais du modèle est de considérer les réponses de l'élève comme un message à transmettre,

² Pour éviter toute ambiguïté, il est important de préciser que les Constraint-Based Tutor, se basent principalement sur des contraintes modélisant le domaine de validité des actions de l'apprenant, c'est-à-dire les contraintes à ne pas transgresser pour éviter les erreurs. Au contraire, dans le cadre du modèle des contraintes, elles sont internes au sujet et décrivent des compétences, des stratégies et des représentations pouvant engendrer ou non des erreurs.

il suffit d'écrire les informations qui indiquent pour chaque problème quelle a été sa réponse parmi les **réponses prédites**. Sans recourir à un modèle, le message aurait une taille plus grande : puisqu'il indiquerait quelle est sa réponse parmi les **réponses possibles**. Soit N_{pred} , le nombre de réponses prédites et N_{poss} le nombre de réponses possibles, la taille en nombre de bits du message compressé serait de $\text{Log}(N_{poss})$ plus petit que $\text{Log}(N_{pred})$, taille du message brut³. Deux extensions sont à faire pour passer de ce cas idéal au cas réel.

Tout d'abord, le nombre de degrés de liberté est à prendre en compte. Avant de décrire le message compressé, il faut indiquer au modèle quel est le profil choisi pour représenter l'élève. Soit $N_{profils}$, le nombre de profils possibles, alors le coût associé est $\text{Log}(N_{profils})$, ce qui correspond au nombre de bits théorique pour décrire le choix du profil. L'élève conserve son profil sur les différents problèmes auxquels il est confronté ; cette quantité est donc comptée une seule fois.

Par ailleurs il faut envisager que le modèle fasse des prédictions incorrectes, et donc penser les cas où les réponses prédites ne contiennent pas la réponse observée. La taille du message serait alors de $\text{Log}(N_{poss}-N_{pred})$. Concrètement, dans l'exemple donné en 1.2.2, le nombre de réponses prédites est de 2 et le nombre de réponses possibles de 8. Si le profil proposé est respecté alors le coût associé est de $\text{Log}(2)=1$ bit, dans le cas contraire il est de $\text{Log}(8-2)=\text{Log}(6)= 2.58$ bits.

Enfin, toujours dans la perspective de transmettre un message représentant le protocole de l'élève, il faut indiquer si la réponse de l'élève est dans le cas 1 (le profil est correct) ou le cas 2 (le profil est incorrect), ce qui coûte un bit supplémentaire par problème. La figure 1 synthétise le calcul de la taille « t » du message compressé.

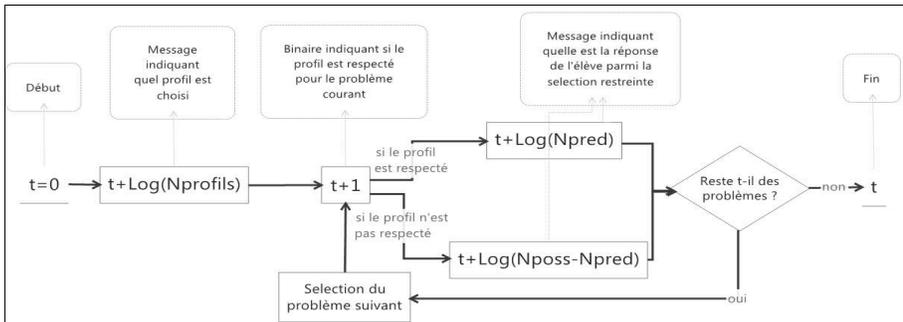


Figure 1. Algorithme de calcul de la taille du message compressé

La longueur totale peut alors être comparée à la taille du message décrivant sans modèle l'ensemble des réponses de l'élève. La taille de ce message est égale à la somme sur l'ensemble des problèmes des $\text{Log}(N_{poss})$ (voir figure 2).

³ Nous avons deux implicites dans nos formulations : tous les logarithmes exprimés dans ce document sont en base 2 (ce qui permet de compter en bits). Même si la dépendance n'est pas marquée pour simplifier l'écriture, N_{poss} dépend du problème considéré, et N_{pred} dépend du profil sélectionné.

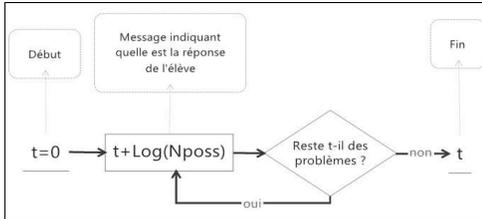


Figure 2. Calcul de la taille du message brut

Un critère permettant de tenir compte et de concilier dans un même calcul le nombre de degrés de liberté du modèle (Nprof), sa précision (Npred), et sa justesse a donc été construit. (voir cependant le 3.4 pour un contrepoint). Il répond à une problématique posée sur le modèle des contraintes, mais est exploitable dans tous les modèles déterministes établissant des prédictions sur un espace d'événements fini.

3.1.3. Utilisation du critère

Le nombre de bits économisés peut être interprété de manière probabiliste. L'existence de 2^k fois plus de messages de 2^N bits que de message de $2^{(N-k)}$ bits implique qu'avec une méthode aléatoire de compression, il y a moins d'une chance sur 2^k d'arriver à une compression d'au moins k bits, ce qui peut donner un point de vue probabiliste intuitif sur la qualité du modèle.

Le critère mis en place peut être utile dans la suite sur deux niveaux : (1) Comme valeur à maximiser : pour chaque élève le profil cognitif qui offre la meilleure compression est sélectionné. (2) Comme mesure de la fiabilité du profil cognitif associé à un élève. Il a donc une place importante dans la modélisation, son calcul est implémenté dans la bibliothèque de fonctions que nous présentons en 3.2.

3.2. Scenarios d'utilisation et description des fonctionnalités de STAR

STAR est composée d'une bibliothèque de fonctions codées en Java destinée au développeur et d'une interface graphique permettant à un intervenant l'accès aux fonctionnalités principales sans qu'il soit professionnel dans le développement informatique.

3.2.1 Introduction

Michèle est une didacticienne intéressée par les apports de la modélisation cognitive pour l'apprentissage en résolution de problème. Elle veut tester et utiliser un modèle décrivant les capacités et expliquant les erreurs de ses élèves dans un domaine sollicitant la résolution de problèmes. Son but est d'affiner puis d'utiliser son modèle dans un EIAH. Elle souhaite établir automatiquement des profils qui lui serviront de base pour une remédiation individualisée. Bien qu'étant relativement à l'aise avec l'ordinateur, elle n'est pas une professionnelle dans le domaine de l'informatique, elle va avoir recours à STAR qui lui facilitera la tâche de modélisation.

3.2.2 Pré requis

Pour tester son modèle dans STAR elle commence par écrire en langage naturel un ensemble de règles constitutives du modèle qu'elle souhaite mettre en place. Cette phase d'écriture préalable lui permet de lister les propriétés des problèmes et les propriétés des réponses qui sont nécessaires pour décrire ces règles. Prenons par exemple, dans le domaine de l'arithmétique⁴, la règle d'interprétation de « d » en valeur absolue que nous avons vu plus haut : Les problèmes qui laissent apparaître une relation de comparaison ont alors la propriété « d de plus » que nous noterons « *Pcomp* », les réponses qui sont compatibles avec cette erreur auront aussi une propriété prévue à cet effet que nous noterons « *Pq2* ». La description de ces propriétés se fait par le biais de tableaux constituant une base de données minimaliste à remplir avant de démarrer STAR.

À ce niveau d'avancement, STAR peut être utilisé pour décrire le modèle, mais pas encore pour travailler avec les réponses obtenues des élèves. Michèle devra donc faire un travail supplémentaire consistant dans un premier temps à **lister les différents types de réponses possibles de chaque problème**, et dans un second temps **indiquer à quel type de réponses possibles s'appartient les différentes réponses des élèves** (aux problèmes auxquels ils ont été confrontés). Ces tableaux peuvent être produits à la main sans difficulté si le nombre d'élève est raisonnable mais s'il s'agit plus tard d'intégrer STAR dans un EIAH, ou de faire des analyses sur une base de données importante, le concours d'un développeur est nécessaire pour se donner la possibilité d'exporter automatiquement les données dans le format requis par STAR. Les manipulations et évaluations du modèle, par contre, sont portées par l'interface de STAR et ne demandent pas de développement informatique supplémentaire.

3.2.3 Utilisation de STAR : manipulation du modèle

Michèle peut maintenant ouvrir STAR pour écrire et tester son modèle. Elle accède à l'onglet **rules** (fig3). Supposons pour la suite qu'elle veuille écrire la règle d'interprétation en valeur absolue telle que nous l'avons décrite plus haut, qu'elle nomme « *constraint2a* ».



Figure 3. Écriture de la contrainte 2a par le biais de l'onglet « rules »

Comme elle a déjà décrit les propriétés nécessaires à l'écriture de cette règle, il lui suffit de d'indiquer au programme que cette règle implique de prendre la deuxième valeur de l'énoncé, mais qu'elle ne s'applique que dans une relation de comparaison où l'expression « d de plus » est présente. La règle s'écrit en langage naturel [*« prendre la deuxième valeur de l'énoncé » SI « relation de comparaison où d de plus est présenté »*] elle rentre le nom de

⁴ Nous rappelons que STAR est un outil générique et que l'arithmétique est choisi dans ce scénario pour conserver le contexte des exemples précédents.

sa règle, la sélectionne, et indique dans le premier panel concernant les propriétés des réponses « *it depends* » ce qui ouvre le second panel à droite concernant les propriétés des problèmes permettant de préciser les conditions pour que la règle soit active.

Une fois qu'elle a construit son ensemble de règles, Michèle passe sur l'onglet « model » (fig 4) permettant d'organiser les règles entre elles. Cet onglet va lui permettre de grouper les règles qui ne peuvent être vraies ensemble, notamment celles qui portent sur les mêmes aspects de la résolution.



Figure 4. Onglet Model

Par exemple, si Michèle a posé une autre règle indiquant que « d de plus » est systématiquement bien interprétée alors elle est en relation d'exclusion avec la règle décrite plus haut car les deux ne peuvent être vraies ensemble. L'onglet model lui permet donc de charger une matrice servant à indiquer ces relations d'exclusion. Des modèles par défaut sont aussi possibles et chargés en un clic : celui où il n'y a aucune relation d'exclusion (« bestCombinaison ») et celui où toutes les règles sont incompatibles entre elles (« bestRule »).

3.2.4 Utilisation de STAR : exploitation et évaluation du modèle

Michèle peut alors tester son modèle dans l'onglet **évaluation** et obtenir les profils sélectionnés de chacun de ses élèves. Lorsqu'elle appuie sur le bouton principal, STAR va chercher pour chaque élève le meilleur agencement de règles (le profil donc) qui le simule au mieux, et indiquer une note sur le pouvoir explicatif du modèle. La quantité mise en jeu, permettant de trouver le meilleur profil et de quantifier la pertinence de ce dernier correspond au calcul du critère décrit précédemment. Michèle peut maintenant à volonté opérer à des modifications sur le modèle, et avoir un feed-back immédiat par l'onglet d'évaluation de la pertinence de ses remaniements.

3.2.5 Scenarios postérieurs

STAR n'est pas seulement un échafaudage permettant d'écrire et tester un modèle. Une fois que le modèle est stabilisé, de multiples voies sont possibles pour l'utilisation automatisée de STAR dans le cadre de la classe. Plusieurs scenarios peuvent être imaginés selon la manière dont STAR est intégré à l'EIAH.

Scenario professeur : Jean est professeur des écoles, STAR a récemment été intégré dans un EIAH qu'il utilise régulièrement, il a donc suivi une formation pour apprendre à manipuler l'interface et pour comprendre la signification des règles constituant les profils sélectionnés. Jean peut directement à partir de STAR ouvrir des données liées à la séance d'exercice de la veille (onglet datas) et déterminer le profil de chaque élève (onglet évaluation) sans avoir besoin de modifier le modèle. Chacun des onglets a des possibilités de sauvegarde, ce qui permet de minimiser les problèmes liés aux fausses manipulations.

Scenario développeur : Marie est une chercheuse en EIAH, elle est intéressée par STAR mais veut pousser plus loin ses fonctionnalités. Elle pourrait par exemple vouloir faire en sorte que les règles prédites puissent changer au cours des interactions dans l'EIAH, ou associer des probabilités aux règles de son modèle. Contrairement à Michèle et Jean, elle ne passera pas par l'interface graphique de STAR, mais par sa bibliothèque de fonctions pour pouvoir l'adapter aux besoins spécifiques conditionnés par l'EIAH sur lequel elle travaille.

4. Conclusion et perspectives

Cet article a soulevé deux questions : comment, dans le contexte d'un EIAH, accélérer le déploiement du modèle des contraintes et mesurer la pertinence des profils qu'il génère ? Pour y répondre, nous avons développé STAR, acronyme de Simple Toolbox to Analyze Reasoning, constitué de plusieurs bibliothèques de fonctions pour mettre en place et évaluer un modèle de règles simple. STAR est doté d'une interface graphique ce qui le rend ainsi accessible à des acteurs non formés à la programmation. Enfin, il implémente le calcul d'un critère construit spécialement pour évaluer la pertinence d'un modèle déterministe.

Cette recherche s'inscrit dans le projet d'intégration du modèle des contraintes dans l'EIAH DIANE [HAKEM 05]. La finalisation de cette incorporation constitue donc la suite logique de nos travaux, permettant alors à DIANE de générer des diagnostics cognitifs basés sur le modèle des contraintes, utilisables dans le cadre d'un tuteur par exemple. Cette intégration permettra d'évaluer la pertinence de STAR de manière plus empirique, par des analyses d'usages et des travaux statistiques portant sur la pertinence des profils obtenus.

Par ailleurs, STAR pourra ensuite évoluer dans plusieurs directions comme l'écriture de modèles probabilistes ou dynamiques.

5. Bibliographie

[ANDERSON 93] Anderson, J. R. *Rules of the mind*. Lawrence Erlbaum. 1993

[ANDERSON 95] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, « Cognitive Tutors: Lessons Learned » *Journal of the Learning Sciences*, vol. 4, no. 2, pp. 167–207, 1995.

[CUMMINS 91] D. D. Cummins, « Children's Interpretations of Arithmetic Word Problems », *Cognition and Instruction*, vol. 8, no. 3, pp. 261–289, 1991.

[GRUNWALD 99] P. Grünwald, « Viewing all models as probabilistic » in *Proceedings of the twelfth annual conference on Computational learning theory*, New York, NY, USA, 1999, pp. 171–182.

[HAKEM 05] Hakem K, Sander E, Labat J-M, J-F Richard « DIANE, a diagnosis system for arithmetical problem solving », AIED 2005, Amsterdam (Pays-Bas), IOS Press, p 258-265

[KINTSCH 85] W. Kintsch and J. G. Greeno, « Understanding and solving word arithmetic problems, » *Psychological Review*, vol. 92, no. 1, pp. 109–129, 1985.

[MITROVIC 01] A. Mitrovic, M. Mayo, P. Suraweera, and B. Martin, « Constraint-Based Tutors: A Success Story » in *Engineering of Intelligent Systems*, L. Monostori, J. Váncza, and M. Ali, Eds. Springer Berlin Heidelberg, 2001, pp. 931–940.

[MYUNG & PITT 05] I. J. Myung and M. A. Pitt, *Advances In Minimum Description Length: Theory And Applications*. MIT Press, 2005.

[OHLSSON 1990] Ohlsson, S., « Why the revolution is not here (yet) », *Cognitive science and instruction*, In H.Mandl, 1990

[RICHARD et al. 93] Richard J. F., Poitrenaud S., and Tijus C., « Problem-solving restructuring: Elimination of implicit constraints » *Cognitive Science*, vol. 17, no. 4, pp. 497–529, 1993.

[RISSANEN 86] Rissanen, J. « Stochastic complexity and modeling. » *The Annals of Statistics* : 1080-1100.1986

[SIMON & NEWELL 71] Simon H. A. and Newell A., « Human problem solving: The state of the theory in 1970 », *American Psychologist*, vol. 26, no. 2, pp. 145–159, 1971.