

ÉVALUATION DES SYSTEMES D'APPRENTISSAGE A BASE DE CONNAISSANCES

SUPPORTING STUDENT REFLECTION IN AN INTELLIGENT TUTORING SYSTEM FOR LOGIC PROGRAMMING

Date: 10-nov.-05

Sommaire

Ce document présente une revue de la littérature au sujet des SABC et détaille l'évaluation d'un Système Tutoriel Intelligent intitulé "Supporting Student Reflection in an Intelligent Tutoring System for Logic Programming".

Référence	Évaluation des systèmes d'apprentissage à base de connaissances (SABC) : "Supporting Student Reflection in an Intelligent Tutoring System for Logic Programming"
Cours	DIC9340 Systèmes d'apprentissage à base de connaissances
Programme	Doctorat en informatique cognitive
Département	Informatique
Institution	Université du Québec à Montréal
Professeur	Roger Nkambou, Ph.D.
Préparé par	Christina BRAZ et Isabelle SAVARD

Note : Dans ce document, le masculin est employé à titre générique afin d'alléger le texte et désigne autant le féminin que le masculin.

TABLE DE MATIÈRES

1	INTRODUCTION	4
2	LES SYSTÈMES D'APPRENTISSAGE À BASE DE CONNAISSANCES	5
2.1	Définition	6
2.2	Les composantes structurelles	6
2.2.1	Le modèle du domaine	7
2.2.2	Le modèle pédagogique	8
2.2.3	Le modèle de l'apprenant	9
2.2.4	L'interface	11
2.3	LA PORTEE DES SABCS	12
3	ÉVALUATION DES SABC	13
3.1	POURQUOI EVALUER LES SABC?	13
3.2	QUOI EVALUER DANS UN SABC?	14
3.3	COMMENT PEUT-ON EVALUER ?	15
3.4	QUAND DOIT-ON EVALUER?	17
3.5	QUI DOIT EVALUER?	21
4	LE SYSTÈME TUTORIEL INTELLIGENT "PROLOG-TUTOR"	21
5	TECHNIQUES UTILISEES POUR L'EVALUATION DE PROLOG-TUTOR	26
5.1	L'évaluation critériée	26
5.1.2	Description	26
5.1.3	Justification du choix	29

6	RÉSULTATS DE L'ÉVALUATION	30
6.1	Évaluation du système selon la grille et présentation des résultats	30
6.1.1	Discussion des résultats de l'évaluation critériée	37
6.2	Évaluation de l'extérieur (« outside assessment »)	39
5	CONCLUSION	42
6	RÉFÉRENCES	42
7	ANNEXES	47

1 INTRODUCTION

Le but de ce document est de présenter l'évaluation d'un système tutoriel intelligent (STI) intitulé "Prolog-Tutor". Ce système est partie intégrante de la thèse de doctorat de Madame Joséphine Tchetagni affiliée au Département d'informatique de l'Université du Québec à Montréal et au Centre de recherche LICEF © Télé-Université sous la direction de Mme Jacqueline Bourdeau, Ph.D. (Professeure au Centre de recherche LICEF © Télé-Université) et de M. Roger Nkambou, Ph.D. (Professeur au département d'informatique et Directeur du Laboratoire Gestion, Diffusion et Acquisition de Connaissances (GDAC) à l'Université du Québec à Montréal).

"Prolog-Tutor" est constitué d'un modèle de dialogue tutoriel qui fournit un guidage systématique à l'apprenant à travers une réflexion explicite et focalisée dans des systèmes tutoriels intelligents (STI). La structure de ce modèle est en fait une interprétation des cinq composantes fondamentales de la "pensée réflexive" selon [Dewey33] dans le champ de l'éducation. Le contenu de ces dialogues se réfère à la nature de l'habileté sur laquelle l'apprenant réfléchit comme déterminé dans la taxonomie d'habiletés.

Les systèmes tutoriels intelligents (STIs) sont des systèmes d'apprentissage un à un (tuteur/apprenant). Ils sont basés sur la méthode du tutorat, qui a été reconnue comme étant la meilleure méthode d'enseignement, et ils ont pour but de reconstituer le comportement d'un tuteur intelligent afin d'offrir un enseignement individualisé à l'apprenant. Les STIs fournissent une possibilité de reproduction dynamique d'exercices, d'ajustements du niveau de difficultés selon les performances de l'apprenant ainsi que d'analyse de l'interprétation du comportement de l'apprenant. Les STIs peuvent produire des inférences sur des connaissances de l'apprenant et interagir de façon intelligente avec lui en ajustant en temps réel les sujets à être exposés en fonction des résultats obtenus et de la formule d'apprentissage qui lui convient le mieux.

Lorsqu'on parle des Systèmes d'Apprentissage à Base de Connaissances (SABC), l'évaluation occupe une place importante et fait partie intégrante du cycle. En effet, plusieurs méthodologies d'évaluation sont disponibles pour différents types d'évaluation et différents critères peuvent être employés afin de mesurer la valeur de ce système. Parmi les aspects positifs en ce qui concerne un STI se trouvent par exemple des composantes telles que la démarche pédagogique adoptée, le type d'interactions proposées à l'apprenant, la capacité d'adaptation du STI aux différences individuelles, son niveau d'ouverture pour des développements futurs et d'autres.

Dans ce document, le terme "apprenant" représente un individu qui suit une formation en ligne. Plus amplement, "il s'agit d'une personne engagée et active dans un processus d'acquisition ou de perfectionnement des connaissances et de leur mise en œuvre" [Afnor96] (e.g. un étudiant qui suit un cours à son institution d'enseignement d'attache, une employée suivant une formation spécifique liée à une entreprise, un infirmier qui consulte un système expert pour diagnostiquer des infections dans un hôpital, etc.).

2. LES SYSTÈMES D'APPRENTISSAGE À BASE DE CONNAISSANCES

Les nombreuses applications des techniques et des principes de l'Intelligence Artificielle¹ à l'enseignement ont générés diverses dénominations pour désigner les produits qui en résultent. Parmi ces dénominations plus usuellement utilisées, nous citons les termes Enseignement Intelligent Assisté par Ordinateur (EIAO) dont les Systèmes Tutoriels Intelligents (STI) et les Systèmes d'apprentissage à Base de Connaissances (SABC) sont partie intégrante et les Environnements Informatiques pour l'Apprentissage Humain (EIAH) [Nkambou05]. Le but de ces systèmes est de simuler l'enseignant dans ses capacités d'expert pédagogue et d'expert du domaine.

"Dans les environnements d'apprentissage, l'accent est davantage mis sur la simulation d'un modèle que sur sa construction. L'apprenant apprend en modifiant des paramètres et en observant les conséquences de ses actions dans l'environnement simulé" [Frasson04].

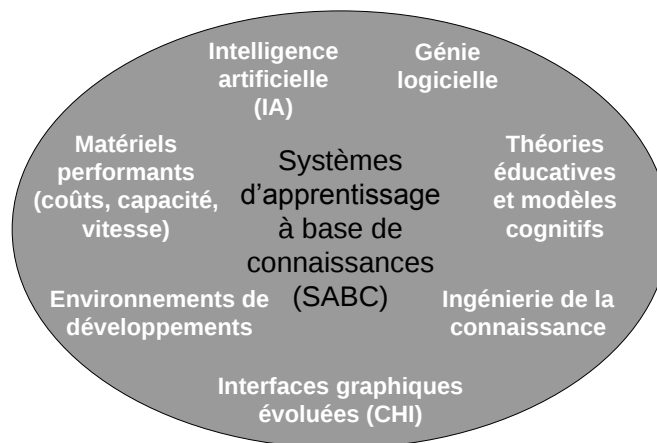


Figure 1: Les domaines impliqués dans les récents développements des SABCs [Nkambou05].

Nous assistons présentement à l'émergence d'environnements, d'outils, de théories d'apprentissage faisant appel à différents domaines (Figure 1). Ceux-ci apportent une contribution importante dans la conception, la mise en œuvre et l'évaluation d'un SABC. Par exemple, dans le domaine des interfaces graphiques évoluées (Computer Human Interaction (CHI)), le but est essentiellement d'éliminer ou de réduire considérablement toute charge mentale² chez l'apprenant.

Généralement les SABC se basent sur des théories d'apprentissage. Parmi les plus connues, mentionnons les théories: i) behavioristes (i.e. stimulus-réponse); ii) cognitivistes (i.e. ces théories du traitement de l'information qui décrivent l'apprentissage comme un ensemble de transformations de l'information qui s'exécutent à travers les structures du cerveau); iii) constructiviste (i.e. théorie qui affirme que que l'apprenant construit son savoir et ses compétences lui-même en fonction de ses

¹ L'intelligence artificielle est définie ici comme étant "l'ensemble des techniques informatiques qui apportent à l'ordinateur une certaine compétence à résoudre des problèmes dans des domaines comme ceux du raisonnement ou de la compréhension du langage naturel" [Schneider98].

² La charge mentale est la partie de la charge cognitive qui est imposée uniquement par une tâche donnée et les demandes environnementales.

expériences et des savoirs déjà acquis [Piaget78]). Autrement dit, "l'individu est un organisme proactif : il ne fait pas que répondre à des stimuli provenant de l'environnement, comme le propose le béhaviorisme, mais s'engage dans une recherche de significations [Perkins91]"; et finalement iv) Socio-constructiviste : cette théorie part du principe « qu'en plus de ce processus de construction individuelle (constructivisme), l'interaction avec des pairs et des experts permet de confronter les schémas cognitifs que l'on a ainsi construit, ce qui développe l'argumentation et la verbalisation et est propice au développement d'une plus grande conscience » [Class&Schneider04].

2.1 Définition

Un système d'apprentissage à base de connaissances (SABC) est un programme bâti pour modéliser les compétences de résolution des problèmes des humains; il est considéré comme un "environnement interactif d'apprentissage" [Nkambou05]. Il existe en effet des systèmes informatiques qui font des déductions et raisonnent sur des connaissances représentant l'expertise du domaine en ayant comme objectif de résoudre un problème, de poser un diagnostic ou de donner un conseil.

Les SABCs possèdent des règles indispensables à l'expert pour engendrer une interaction pédagogique à partir d'un ensemble de connaissances. Ici, les connaissances pédagogiques et celles liées au contenu sont découpées. Lors de la conception du modèle du domaine, il est bon de penser à rendre les connaissances liées au contenu enseigné réutilisables afin de permettre leur utilisation au-delà d'une interaction spécifique.

Quelques exemples des SABC sont « AnimalWatch »³, « Astronomy Tutor »⁴, « OWL (Online Web-based Learning) on-line (Démonstration) »⁵ et plusieurs d'autres.

2.2 Les composantes structurelles

Le déclenchement d'une activité d'enseignement présuppose que le SABC puisse fournir des réponses à quatre questions: Quel contenu enseigner? Comment diagnostiquer les difficultés de l'apprenant? Et quelle méthode ou stratégie d'enseignement adopter ? Quelle la meilleure façon de présenter le contenu à l'apprenant ? Ces questions se réfèrent au modèle du domaine, au modèle de l'apprenant, au modèle pédagogique, et à l'interface utilisateur. L'architecture de base d'un SABC est illustrée dans la Figure 2.

³ <http://ccbit.cs.umass.edu/people/ivon/publications.html>

⁴ <http://ccbit.cs.umass.edu/astro/astro.html>

⁵ <http://owl.cs.umass.edu/owl-c/demo/owldemo.cgi>

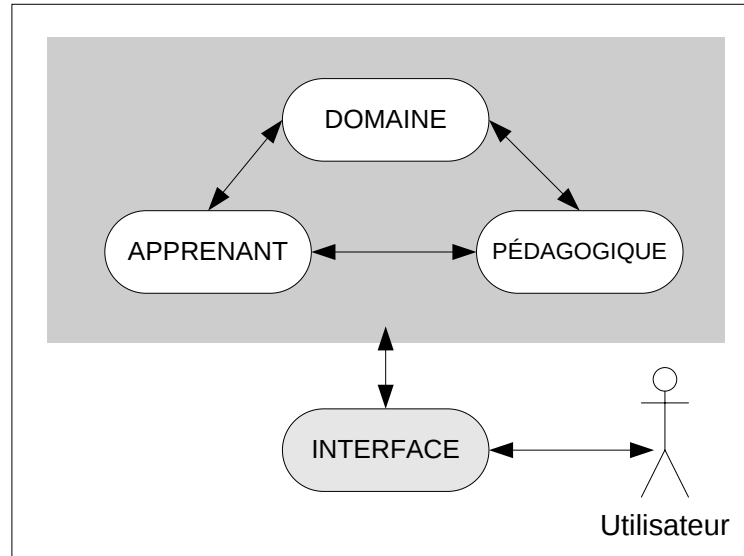


Figure 2: L'architecture de base d'un SABC [Nkambou05].

2.2.1 Le modèle du domaine

Le modèle du domaine se réfère au domaine de connaissance. Il comporte une représentation des connaissances (i.e. le transfert de connaissances d'un expert vers une machine) inhérentes de l'expert en relation au domaine enseigné telles que les faits, les concepts et les stratégies. Cette base de connaissances représente fréquemment 50% de l'effort de développement d'un système d'apprentissage à base de connaissances (SABC) [Nkambou05].

La représentation du domaine n'est pas exclusivement un ensemble de cadres (*frames*) statiques mais effectivement un modèle dynamique du domaine de la connaissance et un ensemble de règles⁶ par lesquelles le système peut raisonner⁷. De ce fait, ce modèle comporte trois possibilités de modélisation:

- (i) un système expert (Figure 3) qui solutionne les problèmes du domaine, qui tient compte d'une variété de possibilités de raisonnement et qui se sert des représentations mentionnées antérieurement en les employant à un cas spécifique;

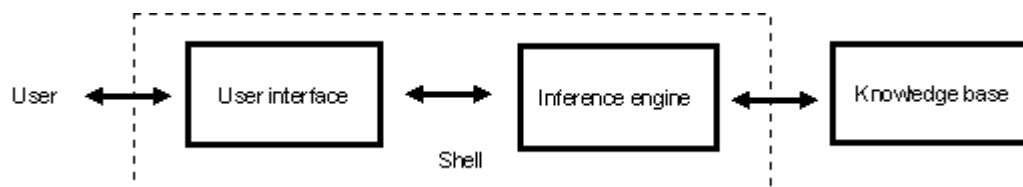


Figure 3: L'architecture d'un système expert [Lurain94].

⁶ Les règles sont des connaissances qui aident à faire la liaison entre des informations connues et d'autres que nous pouvons en déduire ou inférer (e.g. SI un liquide inflammable est déversé ALORS appeler le service d'incendie).

⁷ Le "raisonnement" se réfère au processus de faire coopérer des connaissances, des faits et des stratégies de résolution de problèmes avec l'objectif d'atteindre des conclusions.

Un système expert est composé de trois modules: le module "User interface" (interface utilisateur) représente la communication entre l'utilisateur et le système ; le module "Inference engine" (moteur d'inférence) est un interpréteur pour la base de connaissances. Ce module produit les résultats et les explications pour les problèmes qui lui sont soumis. Les modules "User interface" et "Inference engine" sont alors fréquemment vus comme une seule composante appelé "expert system shell" (coquille de système expert). Le dernier module et la composante principale d'un système expert est alors le module "Knowledge base" (base de connaissances) qui contient les connaissances de résolution de problèmes d'une application particulière.

- (ii) une méthode de raisonnement sur le domaine qui adopte l'approche de type "boîte noire", c'est-à-dire, que nous disposons des entrées, des sorties mais pas des explications, aucun accès à la structure de raisonnement de l'expert. En conséquence, le tuteur devient alors réactif (stimulus-réponse) [Nkambou05];
- (iii) une simulation du comportement de l'humain (e.g. règles que les humains utilisent : heuristiques) employant ses connaissances (modèle cognitif).

2.2.2 Le modèle pédagogique

Un modèle pédagogique sert à instrumenter l'intervention pédagogique du système. Il est constitué d'un ensemble de spécifications sur la façon dont le système doit bâtir ses interventions avec des approches éducatives adéquates. Ce modèle doit pour cela être capable :

- (i) de modéliser l'interaction tuteur ↔ apprenant;
- (ii) de désigner les tâches exposées à l'apprenant d'une façon adaptative;
- (iii) de fournir des rétroactions et de l'aide adaptée à l'apprenant (e.g. le système tuteur pédagogique pourra fournir des rétroactions d'une façon réactive).
- (iv) de mener des activités pédagogiques telles qu'accorder des explications ou des conseils et/ou faire passer des tests.

Selon [Wenger87], "learning is viewed as successive transitions between knowledge states, the purpose of teaching is accordingly to facilitate the student's traversal of the space of knowledge states." Ainsi, un SABC doit identifier les connaissances courantes de l'apprenant et également l'aider dans sa transition vers un nouvel état de connaissance.

Précisons toutefois que la principale finalité de ce modèle est de réduire au minimum les différences de connaissances entre l'expert et l'apprenant. Ainsi, les interventions potentielles que le système pourra entreprendre dans cette dimension sont basées sur différentes stratégies pédagogiques dont les principales sont détaillées ci-dessous :

- (i) "Coaching" : offrir à l'apprenant des conseils et le guider lorsqu'il/elle s'éloigne de la solution [Nkambou05]. Ce mécanisme détermine à quel moment il est convenable d'interrompre l'apprenant pour lui présenter des conseils. Ceux-ci peuvent prendre

- différentes formes telles que : un conseil pour remédier à une situation précise, des renseignements plus détaillés et pourquoi pas des encouragements lorsque l'apprenant répond correctement à une question ou un avertissement lui signalant qu'il/elle n'a pas pris la décision idéale pour un problème donné.
- (ii) Apprentissage par perturbation: un agent artificiel perturbateur, par exemple, peut à la fois fournir des réponses correctes à l'apprenant et parfois essayer de l'induire en erreur en lui donnant des mauvaises réponses afin de vérifier et d'améliorer sa confiance et son estime [Aïmeur&al95], le "troublemaker" [Mengelle&Frasson96].
 - (iii) Apprentissage socratique qui questionne l'apprenant sur ses erreurs pour les faire progresser. Dans [Nkambou05], par exemple, une règle socratique serait :
SI
 L'apprenant donne une explication d'un ou plusieurs facteurs
 qui est insuffisante
ALORS
 Formuler une règle générale pour démontrer que les facteurs
 sont suffisants. Demander à l'étudiant si la règle est vraie.
 - (iv) Le guidage correspondant aux environnements de type micro-monde (e.g. WEST [Brown&Burton79]). Les micro-mondes permettent à l'apprenant de développer et de simuler ses propres modèles du monde réel.

Il est important de souligner qu'il y a des SABC qui fusionnent stratégies tutorielles et modèle du domaine.

2.2.3 Le modèle de l'apprenant

Le modèle de l'apprenant constitue une simulation cognitive de l'apprenant. Il doit permettre un diagnostic et une évaluation de la compréhension du domaine par l'apprenant, de son degré de connaissance et des processus de raisonnement qu'il applique pour une adaptation dynamique et individualisée des interventions du système (e.g. proposer plusieurs niveaux d'exercices et élaborer des rétroactions adéquates pour chaque type d'erreur).

Selon [Nkambou05], ce modèle représente l'état courant des connaissances de l'apprenant et la recherche de cet état s'apparente à un processus de diagnostique. Ce module est consulté régulièrement par les modèles pédagogique et du domaine pour adapter la formation aux besoins de l'apprenant. En plus, il permet d'identifier des connaissances et des compétences émergentes chez l'apprenant.

Pour la réalisation et l'implémentation d'un modèle de l'apprenant dans un SABC, un certain nombre d'approches sont envisageables :

- (i) Le modèle cognitif est fréquemment de type "overlay" (ou modèle de recouvrement) (e.g. tuteur WUSOR [Carr&Goldstein77]). Il est constitué de trois variantes principales :

Overlay de type simple : les connaissances de l'apprentissage sont des sous connaissances de l'expert, donc il y a une comparaison des performances de l'apprenant avec celles de l'expert dans les conditions semblables (Figure 4). Pour cela, il existe plusieurs approches d'implémentation tels que les réseaux sémantiques, l'ensemble de compétences acquises par l'apprenant et les réseaux bayesiens.

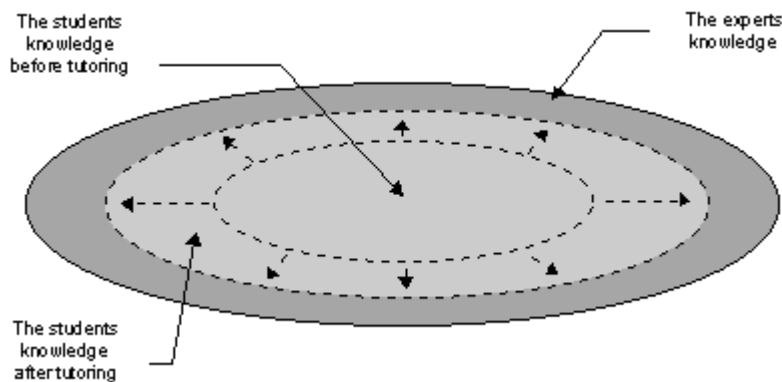


Figure 4: La représentation d'un modèle cognitif de type "overlay" illustrant les effets du tutorat [Smith98].

Overlay de type différentiel : il est une extension du modèle overlay (e.g. WEST [Brown&Burton79]). Ce modèle est divisé en ce à quoi l'apprenant est exposé et ce à quoi il n'est pas exposé (Figure 5). En fait, il partitionne les connaissances du domaine en connaissances déjà connues et celles qui ne sont pas encore présentées à l'apprenant.

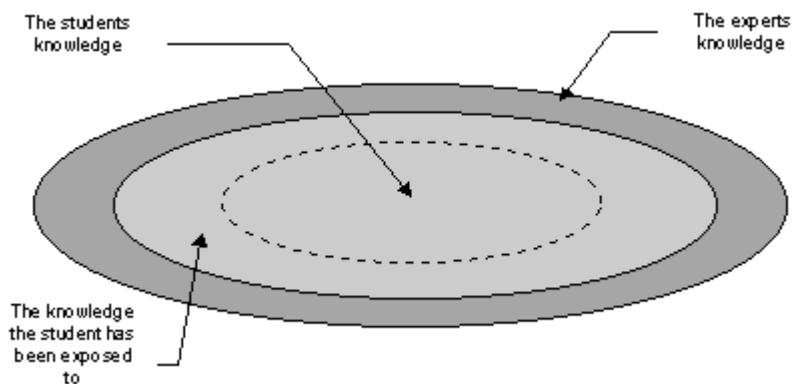


Figure 5 : La représentation d'un modèle différentiel [Smith98].

Overlay de type perturbateur (buggy en anglais) : ce modèle représente les erreurs de raisonnement de l'apprenant constatés lors des étapes d'interaction (Figure 6) (e.g. BUGGY model [Brown&Burton78]). Il présente les connaissances tenues par l'apprenant qui ne figurent pas dans les connaissances du domaine tuteur. "... sometimes known as *mal-rules* (bad rules) that describe student misconceptions in relationship to the experts' knowledge" [Smith98]. Son objectif est de comprendre l'origine de l'erreur et de proposer une stratégie pour remédier à la situation.

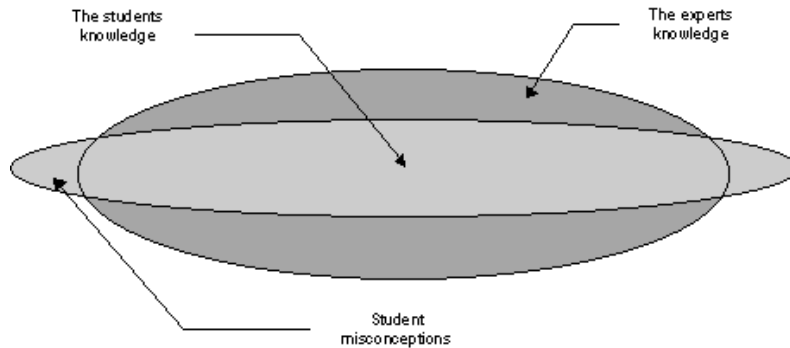


Figure 6 : La représentation d'un modèle de l'apprenant de type perturbateur [Smith98].

- (ii) Le modèle affectif permet le stockage des préférences de l'étudiant, de son profil psychologique, de son profil émotionnel, etc. Ces informations peuvent varier selon le contexte (e.g. motivation) et peuvent aussi évoluer dynamiquement (e.g. état émotionnel). Le tuteur utilise ces informations pour adapter l'interaction avec l'étudiant [Nkambou05].
- (iii) Le modèle inférentiel se réfère essentiellement au diagnostic par l'inférence en relation aux causes des erreurs de l'apprenant tels que "Model Tracing" [Anderson83] (i.e. suivre et analyser la trace des activités de l'apprenant) et "Knowledge Tracing" (i.e. analyser un événement d'apprentissage afin de déterminer les connaissances qui ont été employées).

Lorsqu'on conçoit des SABC, on doit toujours garder en tête qu'ils sont destinés à des humains et qu'il importe de dépasser la puissance de la technologie et de réfléchir aux derniers développements des recherches dans le domaine des sciences cognitives. Ce n'est pas l'ordinateur, ni le programme qui "jouent le rôle essentiel" mais bien l'apprenant. Sans sa participation active, sans sa motivation, son envie d'apprendre, aucun logiciel ne pourra lui transmettre des connaissances de manière significative [Retschitzki95].

2.2.4 L'interface

L'interface est basée sur des principes d'ergonomie de l'interaction, c'est-à-dire, qu'elle sert à faciliter la communication entre le système et l'utilisateur. De nos jours, les logiciels fournissent une interface graphique très raffinée, les managements directs (i.e. les interactions) y sont primordiaux, ainsi l'écran lié à la souris collaborent à la construction du savoir de l'apprenant.

Les aspects pédagogiques d'une interface dans un SABC doivent être pris en compte autant que les aspects d'interaction mentionnés ci-dessus. Dans [Wenger87], l'objectif de la communication de connaissances ("knowledge communication") requiert qu'une interface soit composée d'un modèle de discours afin de résoudre les ambiguïtés concernant les réponses de l'apprenant (i.e. attendu que l'apprenant peut fournir des réponses incomplètes et contradictoires dans un moment de blocage, une adéquate rétroaction du système peut s'avérer fondamentale) (e.g. SOPHIE). Un autre aspect de l'interface se réfère à la présentation des connaissances ("knowledge presentation"). La mise en disponibilité des connaissances à travers l'interface (e.g. World Wide Web) ne suffit pas à elle seule pour communiquer les connaissances ("knowledge communication"). Le SABC doit fournir du "coaching" ou du guidage afin d'éviter que l'apprenant ne se décourage ou qu'il néglige des aspects importants du domaine.

Dans un contexte d'apprentissage [Dufresne01] souligne qu'il est important d'établir un dialogue entre le système d'apprentissage et l'apprenant, de façon à ce que le contrôle soit partagé, "pour que l'apprenant puisse approcher la matière à son rythme en fonction de sa compréhension et que des mécanismes de surveillance et de soutien puissent le guider et l'appuyer dans sa démarche". Tout cela peut nous rappeler le modèle pédagogique!

2.3 La portée des SABCs

Le fondement des SABCs consiste à rassembler le maximum d'informations originaires du savoir d'un domaine (e.g. la construction d'une base de connaissances) dans un système automatisé permettant un accès partagé. Ainsi, pour que nous puissions bâtir ce système, une combinaison des experts de différents domaines tels que l'éducation, les sciences cognitives, la psychologie, l'ergonomie, l'informatique et d'autres contribuant à la conception, au développement et à la fabrication des SABCs est donc requise. Partant de cette caractéristique multidisciplinaire, les SABCs permettent une plus grande vitesse des prises de décisions ainsi qu'une fréquence augmentée des révisions, et en outre offre une solution pour des problèmes d'acquisition des connaissances (e.g. préservation de l'expertise d'une personne).

Comme nous l'avons vu précédemment, les SABCs incorporent l'expertise humaine pour faciliter le processus de prise de décision et la résolution idéale de divers problèmes plus complexes. Ils sont constitués de logiciels et d'ordinateurs qu'intègrent des sous-systèmes plus raffinés. Ces technologies reproduisent et accroissent la capacité des humains à percevoir, à raisonner, à prendre des décisions et également à poser des actions. Un autre facteur plus tangible de l'utilité réelle des systèmes experts serait par exemple la diminution de la charge cognitive sur certains aspects de la tâche de raisonnement.

Les SABCs en effet permettent aux ordinateurs de devancer les requêtes et de traiter avec des environnements qui sont complexes, inconnus et dynamiques. Ces systèmes comprennent les concepts de reconnaissance statistique de formes, les réseaux de neurones, le raisonnement par cas, la logique floue, les algorithmes génétiques et l'exploration de données (Data Mining). Ce domaine de recherche comporte également les travaux liés aux bases de données, aux agents, aux systèmes de raisonnement et aux agents intelligents.

Enfin, les SABCs permettent l'organisation de connaissances, l'aide au travail courant et la diffusion des savoirs et spécialement la formation.

Les SABCs ont été employés à l'enseignement, au diagnostic d'équipement, à la planification de procédés et à l'identification des usagers. Ces systèmes rencontrent en outre des applications variées dans nombreux secteurs de l'industrie tels que l'enseignement (e.g. modélisation de l'apprenant : BUGGY model [Brown&Burton78] ; mathématique: WEST [Brown&Burton79]), l'agriculture (e.g. aide à la décision pour la production animale) : [Huirne&Dijkhuizen88]), la fabrication (e.g. simulation d'une chaudière de récupération : RBT (Recovery Boiler Tutor) [Woolf86]), le transport (e.g. analyses de sécurité routière: [Boury&Tourigny97]), les communications (e.g. gestion des réseaux : GISBC [Bellafkih&al01]), l'analyse financière (e.g. aide à la décision et systèmes experts : [Ernst87]), la production d'énergie électrique (e.g. diagnostic de panne électronique : SOPHIE [Brown&Burton74]), et la santé (e.g. diagnostic médical : GUIDON [Clancey82]).

Autrement dit, nous revendiquons aussi bien que [Lurain94] que la puissance de la technologie informatique appliquée à l'enseignement ne suffira pas à elle seule à créer les meilleures SABCs (ou au moins en essayant de s'approcher de la simulation du comportement de l'humain). Il faudra également que les théories cognitives évoluent, pas exclusivement pour décrire comment nous interagissons avec ces systèmes, mais pour accommoder les changements que ces systèmes apporteront aux processus cognitifs. En paraphrasant l'ergonomie cognitive, cela est une double-tâche⁸ à être réalisée par les partisans de l'intelligence artificielle aussi bien que de l'éducation.

3 ÉVALUATION DES SABC

Selon [Cooley&Lohnes76], l'évaluation est le processus qui permet de recueillir des données pertinentes et de les transformer en informations pour la prise de décision. [Mark &Greer92] nous rappellent que le développement d'une évaluation est un processus complexe et que les Systèmes Tutoriels Intelligents sont eux-mêmes complexes. Les quatre composantes d'un STI (l'Interface, le modèle du domaine, le modèle pédagogique et le modèle apprenants) ne s'observent pas nécessairement sous la même loupe. Les auteurs suggèrent qu'un grand nombre de critères soient utilisés pour évaluer de tels systèmes. Généralement, plus d'une technique d'évaluation doit également être utilisée. [Nogry&al02] en ont d'ailleurs fait l'objet d'un article intitulé « Évaluation des EIAH : une nécessaire diversité des méthodes ».

3.1 Pourquoi évaluer les SABC?

L'économie de temps et d'argent tout en augmentant la qualité et l'efficacité des systèmes sont toujours une préoccupation pour les équipes de conception de SABC. L'évaluation permet généralement d'atteindre ces objectifs.

⁸ Le paradigme de la "double-tâche" (ou tâche ajoutée) en Ergonomie cognitive se réfère au partage des ressources attentionnelles impliquant un certain degré de concurrence imposé entre deux ou plusieurs tâches : une tâche primaire est maintenue constante pendant qu'une tâche secondaire requiert occasionnellement l'attention. (e.g. une tâche principale : opérations de calcul) et une tâche secondaire (mémorisation d'une liste de mots)).

Selon [Nogry&al02] il est maintenant bien établi que l'évaluation des EIAH est essentielle. On peut évaluer pour s'assurer que le système en développement répond bien à des besoins d'apprentissage, qu'il peut être utilisé de façon efficace par la clientèle cible, que le modèle du domaine est complet et juste, que les techniques pour recueillir de l'information au sujet de l'apprenant permettent effectivement de construire un modèle de l'apprenant.

[Mark &Greer92] définissent deux grands types d'évaluation : l'évaluation formative et l'évaluation sommative. L'évaluation formative sert à recueillir des informations dans le but de modifier un système, d'en corriger certains aspects, de l'améliorer. L'évaluation sommative sert plutôt à sanctionner, à se prononcer sur un ou plusieurs aspects d'un système qu'on dit complété. Par exemple, on pourra évaluer, de façon sommative, un système dans le but de décider s'il sera acheté ou non. L'évaluation sommative nous semble près de l'approche quantitative décrite par [Iqbal&al99]. Et l'évaluation formative plus près de leur approche qualitative.

Nous avons décidé de mener une évaluation formative dans le but de permettre à la conceptrice de recueillir des informations qui lui permettront d'améliorer son système qui n'est encore qu'un prototype.

3.2 Quoi évaluer dans un SABC?

On peut évaluer la conformité, c'est-à-dire vérifier si les fonctions pédagogiques spécifiées et décrites sont vraiment réalisées. On peut vérifier l'efficacité, c'est-à-dire vérifier si la tâche pédagogique peut être réalisée dans un délai raisonnable. On peut aussi se préoccuper de l'analyse et de la conception pédagogique et chercher à vérifier l'utilité du système, à s'assurer que des besoins d'apprentissages sont effectivement comblés. Dans le cadre de ce travail, nous parlerons d'avantage d'utilisabilité.

Utilisabilité

Lorsqu'on évalue l'utilisabilité d'un système, on cherche à savoir si l'interface permet une interaction facile pour l'utilisateur principal mais aussi si le système favorise l'acquisition de connaissances, s'il peut être utilisé efficacement dans différents contextes (individuellement, en groupe, en équipes, etc.) et si son utilisation permet de maintenir ou d'augmenter la motivation de l'apprenant.

Pour présenter les spécificités liées à l'évaluation de l'utilisabilité des EABC, Nogry et al reprennent la distinction proposée par Senach entre évaluation analytique et évaluation empirique.

Évaluation analytique

Selon Nogry et al., l'évaluation analytique consiste à étudier les interface selon un ensemble de référents afin de contrôler qu'elles possèdent bien certaines qualités et de détecter les problèmes qu'elles peuvent poser. Ce type d'évaluation est souvent pratiqué à l'aide de listes de critères élaborées par des ergonomes. Une autre façon de pratiquer l'évaluation analytique pourrait être de procéder à l'aide de cas d'utilisation. Les cas d'utilisation permettent d'imaginer toutes les utilisations (tâches et séquences d'action) possibles du système par les apprenants.

Ce type d'évaluation n'implique pas d'apprenants représentatifs de la clientèle cible. Notre évaluation critériée se base principalement sur les travaux de Nielson.

Évaluation empirique

L'évaluation empirique permet de recueillir des données au sujet du comportement des utilisateurs. Selon Nielsen, cinq utilisateurs représentatifs permettent de déceler 80% des problèmes d'utilisation du système. Comme nous n'avons pas accès aux apprenants et comme nous n'avons pas facilement accès à Tutor-Prolog non plus, nous n'avons pas opté pour ce type d'évaluation. Mais, nous savons qu'un aspect de Prolog-Tutor a été évalué de cette façon selon l'hypothèse de recherche suivante :

Hypothèse de recherche : l'intégration de la dimension de la réflexion confère un caractère pédagogique plus manifeste au Diagnostic Cognitif (DC).

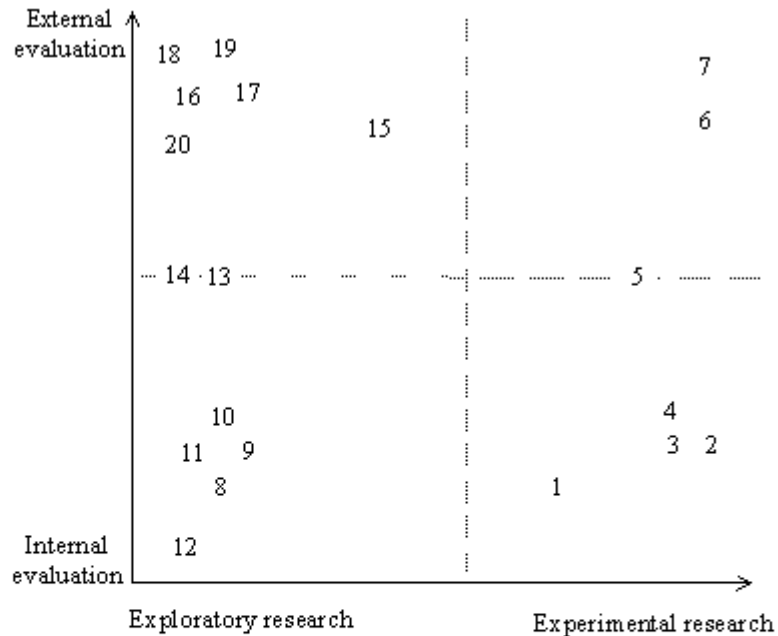
Six apprenants représentatifs ont été observés pendant qu'ils interagissaient, en pensant à voix haute, avec Prolog-Tutor.

3.3 Comment peut-on évaluer ?

Deux grandes approches : l'approche expérimentale et l'approche qualitative

[Ikbal&al99] proposent une méthodologie d'évaluation à deux dimensions. La première dimension touche l'étendue des aspects évalués et permet la distinction entre l'évaluation interne, qui vérifie un aspect ou d'une partie du système, et l'évaluation externe, qui vérifie le système entier. La deuxième dimension distingue des approches particulières d'évaluation.

Deux grandes approches sont alors définies : l'approche expérimentale et l'approche qualitative (ou exploratoire). La première réfère à des méthodes systématiques et rigoureuses d'évaluation et elle nécessite un grand nombre de participants qui sont généralement séparés en deux groupes, le groupe expérimental et le groupe contrôle. La deuxième est adoptée lorsqu'on cherche à évaluer un phénomène particulier relié à l'apprentissage et/ou qu'on dispose d'un nombre limité d'apprenants. Au total, [Ikbal&al99] distinguent vingt méthodes d'évaluation qu'ils ont toutes situées selon la Figure 7 :



Key to evaluation methods:

- | | |
|--|--|
| 1. Proof of Correctness | 2. Additive experimental design |
| 3. Diagnostic accuracy | 4. Feedback/instruction quality |
| 5. Sensitivity Analysis | 6. Experimental research |
| 7. Product evaluation | 8. Expert knowledge |
| 9. Level of agreement | 10. Wizard of Oz experiment |
| 11. Performance metrics | 12. Internal evaluation |
| 13. Criterion-based | 14. Pilot testing |
| 15. Certification | 16. Outside assessment |
| 17. Existence proofs | 18. Observation & qualitative classification |
| 19. Structured tasks & quantitative classification | 20. Comparison studies |

Figure 7: Diagramme de classification des méthodes d'évaluation [Iqbal&al99].

Différentes techniques

Les techniques de l'**approche expérimentale** sont au nombre de sept :

1. La preuve de véracité permet d'évaluer la correspondance entre l'architecture d'un système, son comportement et ses spécifications. On vérifie si le système fait bien ce qu'il devait faire. Cette technique n'est pas vraiment appropriée pour les STI en raison de leur complexité et du caractère adaptable.
2. Le Design expérimental additif.
3. Le Diagnostic de l'exactitude
4. La qualité des rétroactions et des instructions
5. L'analyse de la sensibilité

6. La recherche expérimentale
7. L'évaluation du produit

Les techniques de l'**approche qualitative** sont au nombre de treize :

1. Les connaissances de l'expert
2. Le niveau d'acceptation
3. L'expérimentation « Wizard of Oz »
4. Les mesures de performances
5. L'évaluation interne
6. L'évaluation critériée
7. Les tests pilotes
8. La certification
9. L'évaluation de l'extérieur
10. Les preuves d'existence
11. L'observation et la classification qualitative
12. Les tâches structurées et l'observation quantitative
13. Les études de comparaison

Notre approche d'évaluation est qualitative car nous cherchons à évaluer le système sous des aspects particuliers, qui seront décrits plus loin dans ce texte, et nous disposons d'un nombre très limité d'apprenants. Trois des techniques énumérées par [Iqbal&al99] seront utilisées pour l'évaluation de « Prolog-Tutor »: l'évaluation critériée (13), les tests pilotes (14) et l'évaluation de l'extérieur (16).

3.4 Quand doit-on évaluer?

La question du « quand » évaluer a fait couler beaucoup d'encre dans le domaine de la technologie éducative. Il y a eu une époque où les systèmes n'étaient évalués que lorsqu'ils étaient terminés. On invoquait le manque d'argent, de temps, d'outils pour éviter de mener des évaluations en cours de développement. Les propos de [Dick87] sont concluant à cet égard « The argument most often heard for not doing formative evaluation are lack of time and resulting increase in costs if it is done. » Peu à peu l'évaluation formative, en cours de développement, a intéressé plus largement la communauté scientifique. En 1984, [Marton&Harvey 94] parlaient de trois moments privilégiés pour évaluer:

avant, pendant et après le développement. L'évaluation formative menée « après » était alors beaucoup plus appliquée. Avec l'avènement des Systèmes d'Apprentissages Multimédiatisés Interactifs (SAMI), les gestionnaires de projets se sont intéressés plus sérieusement aux évaluations formatives en cours de développement. Comme le développement de tels systèmes exigeait la mise sur pied d'une équipe pluridisciplinaire et l'utilisation de matériel coûteux, on a cherché à s'assurer que le SAMI réponde aux besoins exprimés, qu'il soit intéressant et surtout, qu'il soit utilisé.

Aujourd'hui, les bénéfices de l'évaluation formative en cours de développement de systèmes complexes sont bien reconnus. Il est admis qu'on peut sauver temps et argent en évaluant à différentes étapes du processus. Mais il peut être intéressant de repenser aux différents types de cycles de développement qui sont entrés dans la production de logiciels au fil de l'histoire et à la place qu'a eu l'évaluation dans chacun de ces cycles :

- 1) Le cycle de développement en cascade où l'évaluation avait une place vers la fin de la cascade (Figure 8).

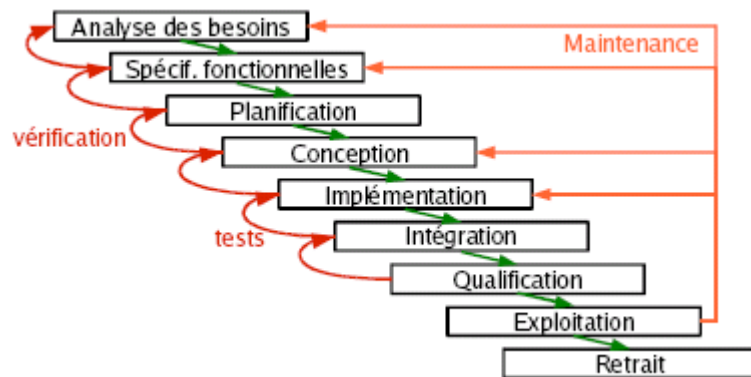


Figure 8 : Cycle de développement en cascade [Marlet05].

- 2) Le cycle de développement en V dans lequel on avait fait une plus grande place à l'évaluation dans le but de limiter les retours aux étapes précédentes (Figure 9).

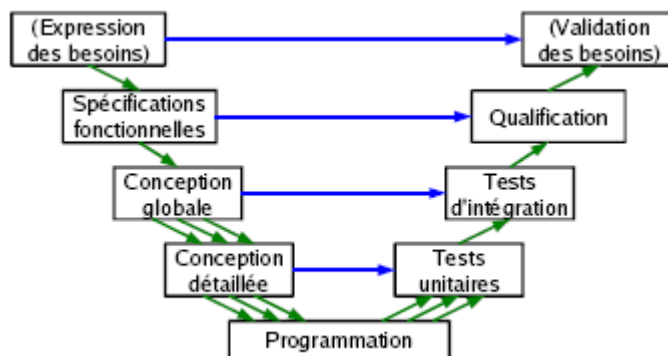


Figure 9 : Cycle de développement en V [Marlet05].

- 3) Le cycle de développement en spirale qui, par l'implémentation de versions évaluées successives, présente un produit de plus en plus complet (Figure 10).

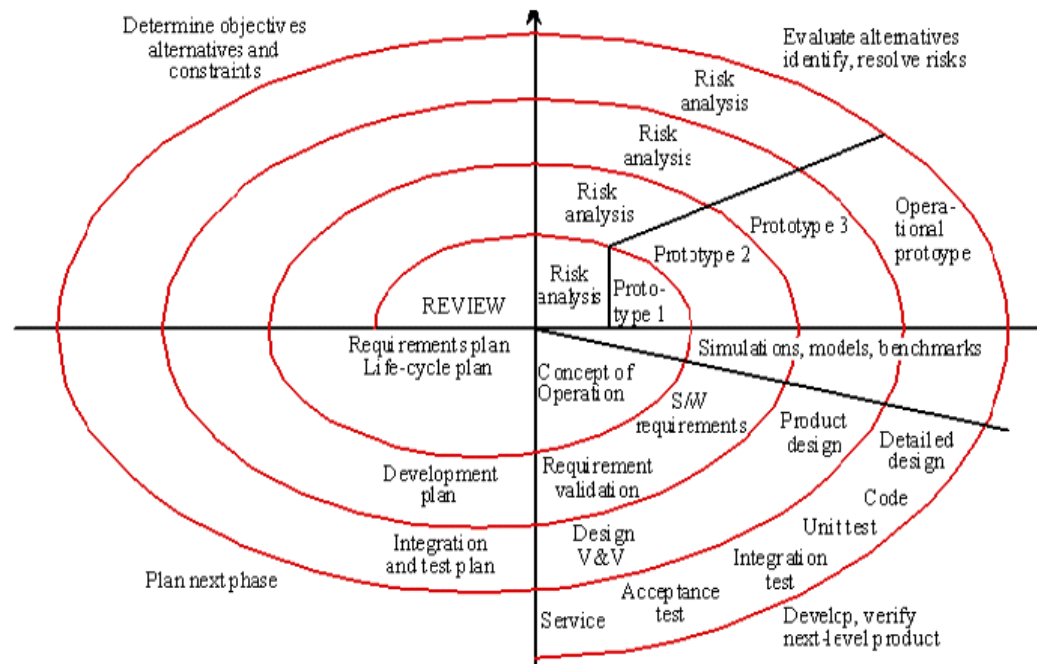


Figure 10: Cycle de développement en spirale [Boehm81].

- 4) Le cycle itératif qui permet de livrer le plus tôt possible un produit qui puisse être testé par des représentants de la clientèle cible (Figure 11).

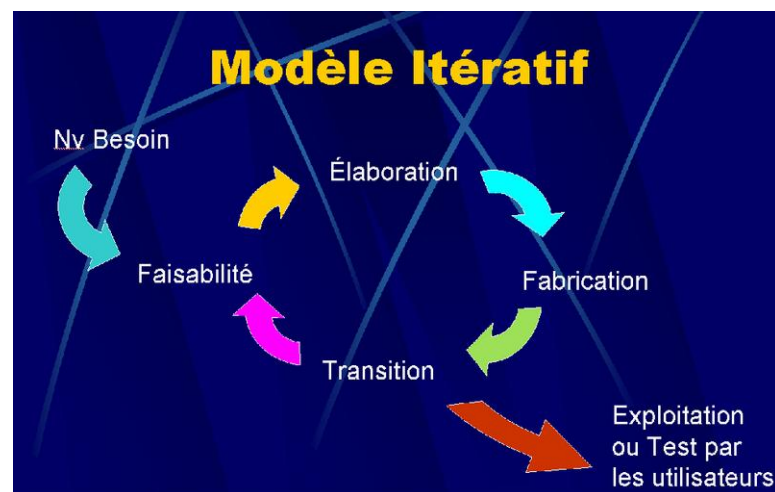


Figure 11 : Les phases du Cycle itératif [Wikipédia05].

Le « Rational Unified Process » (RUP)

Le RUP [Galic&al04] est un processus itératif, à différentes étapes. Qui a pour but d'assurer la production de logiciels de haut niveau qui satisfasse les besoins des utilisateurs dans une programmation et budget prévisibles.

L'architecture de RUP

Une vue générale de l'architecture de RUP est présentée dans la Figure 12 Le processus est composé de deux dimensions selon ci-dessous :

- La dimension horizontale qui représente le temps et indique les phases et l'itération des événements marquants qui se produisent durant le cycle de vie du projet.
- La dimension verticale qui représente le contenu en groupes logiques appelés "disciplines".

Les "bosses" représentées sur le diagramme indiquent l'accent mis sur les disciplines dans le temps. Le développement itératif est indiqué et nous pouvons bien voir toutes les disciplines représentées dans chacune des itérations.

La croix noire indique la phase à laquelle se trouve le système tutoriel intelligent Prolog-Tutor présentement. Prolog-Tutor est encore un prototype non-terminé.

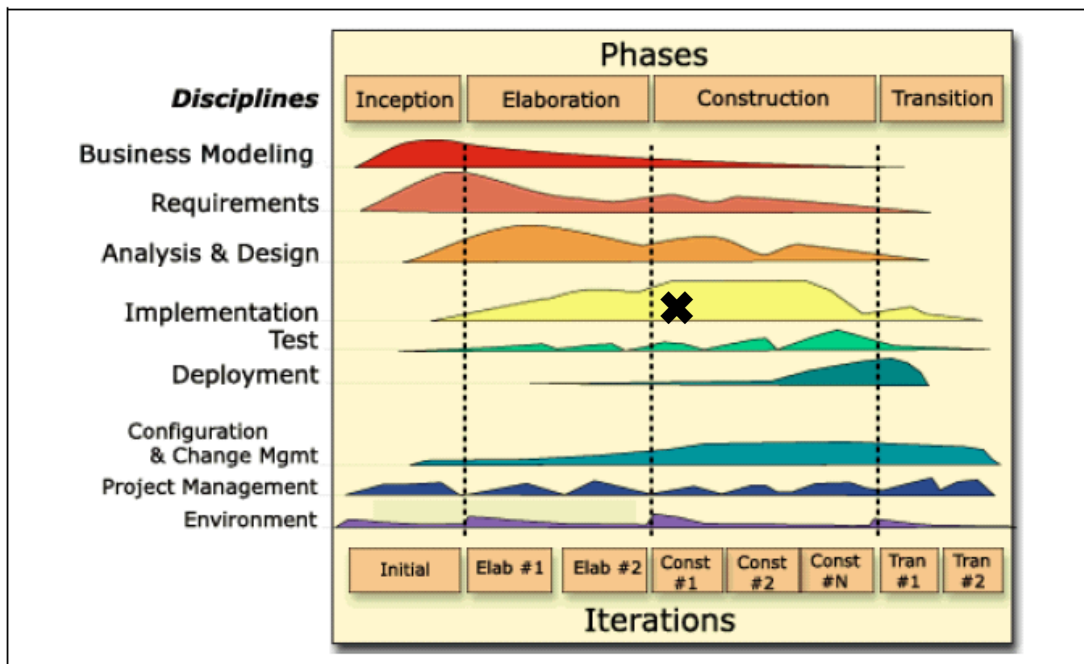


Figure 12: L'architecture de RUP [Galic&al04].

Les phases

Un concept très important dans le RUP est le concept de phases. Elles fournissent les événements marquants ("milestones") qui assurent le progrès et convergent vers une solution plutôt qu'itérer indéfiniment. Les phases de RUP sont respectivement :

- Élaboration : l'architecture du logiciel est établie et validée avec une version exécutable du système.
- Construction: l'accent est mis sur l'achèvement du développement du système.
- Transition : le logiciel est déployé et accepté par ses utilisateurs finaux.

Les objectifs de chacune des phases sont atteints par l'exécution d'une ou plusieurs itérations dans la phase. Chacune des phases se réalise avec un événement marquant qui permet de déterminer si le projet a atteint les objectifs de la phase. Le projet ne peut avancer à la prochaine phase que si les objectifs visés sont atteints.

3.5 Qui doit évaluer?

Différentes personnes peuvent être consultées pour évaluer des SABC. Par exemple, on aller chercher l'avis des experts pour construire le modèle du domaine et/ou en vérifier le contenu, à des sujets représentatifs de la clientèle cible pour vérifier la convivialité, la facilité de navigation, l'efficacité du système ou autre. L'évaluation peut être menée de l'intérieur de l'équipe de conception ou de l'extérieur. [Savard96] rappelle que, selon le cas, les buts d'évaluation peuvent varier et, forcément, les résultats aussi.

4 LE SYSTÈME TUTORIEL INTELLIGENT "PROLOG-TUTOR"

Cette section décrit les caractéristiques du système Prolog-Tutor.

Description du système

Prolog-Tutor est un Système Tutoriel Intelligent(STI) dans lequel l'apprenant est guidé, au moyen de « dialogues tutoriels », dans son apprentissage de la programmation logique.

Clientèle cible

Les étudiants de premier cycle en Informatique.

Objectifs pédagogiques visés

- Identifier et utiliser correctement les structures de base (variables, constantes, termes composés, faits, règles Prolog, base de données, etc.).
- Comprendre et appliquer le principe relié à l'association d'une variable (comprendre et appliquer l'unification) :
 - Comprendre l'unification (concept)
 - Appliquer l'unification (procédure)
 - Comprendre la résolution (concept)
 - Appliquer la résolution (procédure)

Interface

L'interface est divisé en cinq zones spécifiques (Figure 13) : une pour l'énoncé des problèmes, une pour la base de connaissance Prolog, une pour les outils, une pour les dialogues système-usagers et la dernière pour les outils qui permettent à l'apprenant de dialoguer (de répondre aux questions).

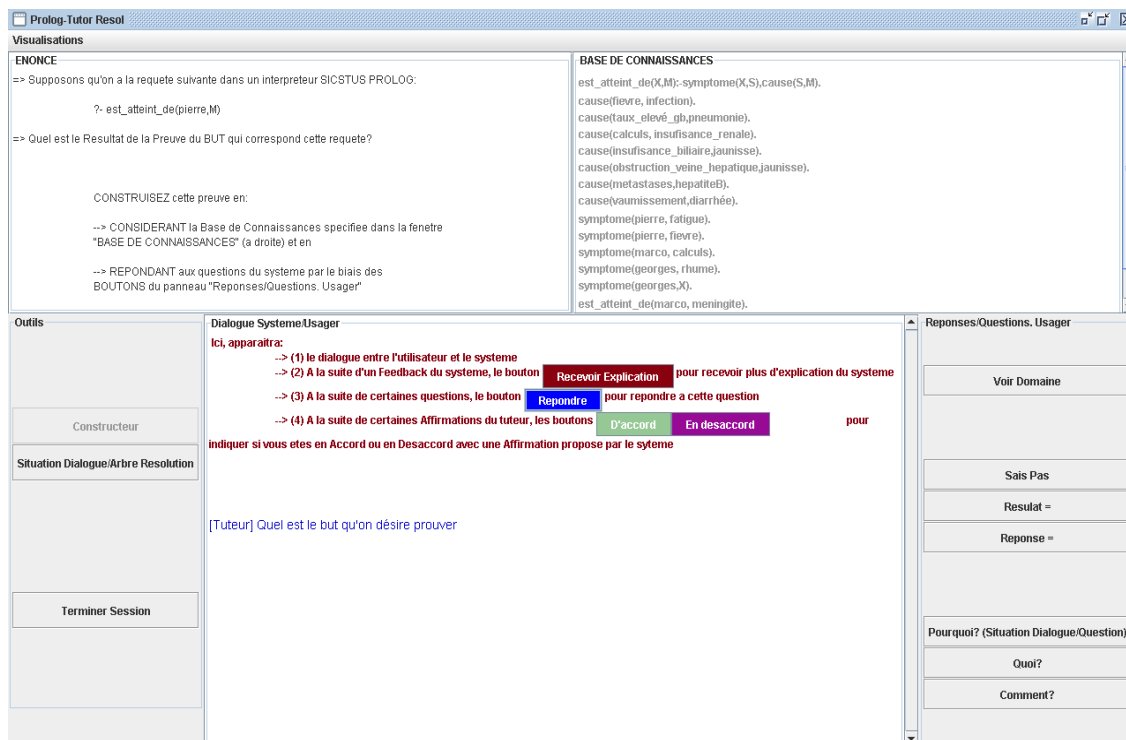


Figure 13 : Interface principale de Prolog-Tutor [Tchetagni&al05].

Modèle pédagogique

La stratégie pédagogique choisie est celle du « dialogue tutoriel » qui guide l'apprenant en l'amenant à réfléchir en profondeur et de façon explicite au sujet de sa démarche d'apprentissage. La structure de ce modèle (Figure 14) est une interprétation des composantes fondamentales de la pensée réflexive telle que définie par [Dewey33]. Ce que Dewey entendait par pensée réflexive était « une manière de penser consciente de ses causes et de ses conséquences »⁹.

Composantes fondamentales de la pensée réflexive

1. Une expérience, contenant une situation obscure/ contradictoire, dans laquelle l'apprenant doit raisonner afin de trouver une conclusion ou solution acceptable.
2. Un processus d'intellectualisation, déclenché par la situation contradictoire, qui consiste en l'identification formelle des conditions qui causent la contradiction.
3. Le processus de réflexion qui consiste en l'analyse (pondering upon) des faits d'une situation dans le but de trouver une conclusion acceptable.
4. L'apprenant réflexif peut aussi appliquer un processus de réflexion interne qui l'amène à relier les idées de façon à inférer des hypothèses supplémentaires.
5. Le résultat du raisonnement, après considération des hypothèses, qui mène à la sélection d'une seule conclusion.

⁹ Pallascio, R., Daniel, M-F., Lafortune, L (2004) « Une pensée réflexive pour l'éducation », Presses de l'Université du Québec. Sainte-Foy (Québec).

Dans le but de respecter les différents points de vue sous lequel un élément de contenu peut être abordé, les éléments de contenu ont tous été liés à des objectifs pédagogiques. Prolog – Tutor distingue donc les éléments de contenu à partir des objectifs auxquels ils ont été associés. On peut dire que les objectifs pédagogiques jouent le rôle de connecteur.

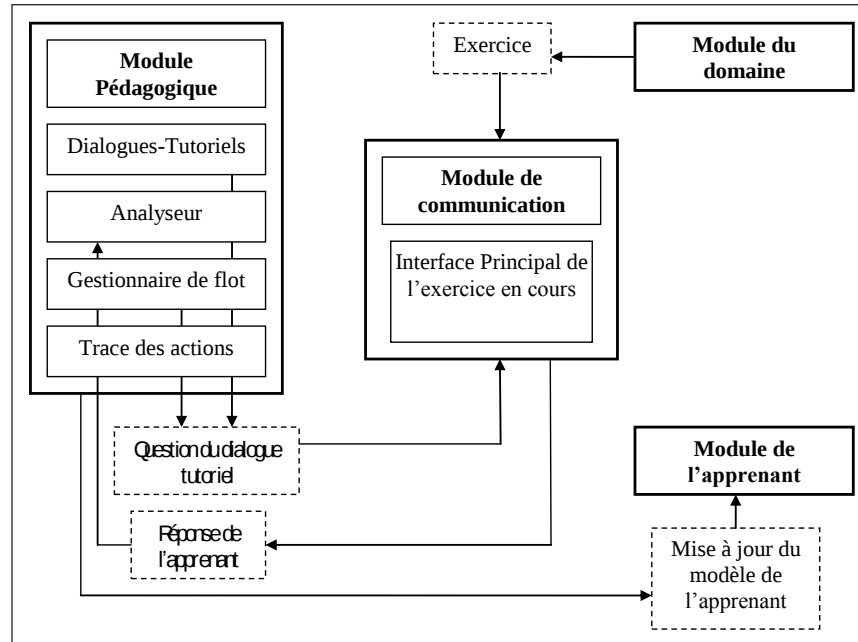


Figure 14 : Architecture du modèle pédagogique de Prolog-Tutor [Tchetagni&al05].

Lorsqu'un apprenant donne une réponse erronée à une question, le tuteur intervient. La stratégie d'intervention du tuteur est de décomposer un problème en sous problèmes. Ces sous problèmes sont décomposés au moyen d'un dialogue avec l'apprenant dans lequel chaque question est associée à une compétence particulière. Cette façon de faire permet au tuteur de faire son diagnostic cognitif en s'appuyant sur une trace tangible. Les objectifs pédagogiques sont utilisés par le modèle pédagogique uniquement pour pouvoir mettre à jour le modèle de l'apprenant.

Le diagnostic cognitif est le processus qui permet de faire des inférences au sujet de l'état cognitif de l'apprenant. Il permet de recueillir des informations au sujet des connaissances que possède ou non l'apprenant mais aussi au sujet de sa façon de raisonner, de ses réflexions à propos du problème à résoudre. Ces informations sont utiles car elles permettent au système d'enregistrer la trace de l'apprenant et ainsi de s'adapter à ses besoins, elles permettent aussi à l'apprenant de s'arrêter pour réfléchir aux connaissances acquises et à leur utilité et ainsi d'apprendre à apprendre.

L'apprenant est toujours questionné de façon à favoriser la pensée réflexive, c'est-à-dire de façon à l'amener à réfléchir au sujet de ses stratégies d'apprentissage, de réflexion.

Le modèle de l'apprenant est un modèle de type Overlay (Figure 15), c'est-à-dire que les connaissances de l'apprenant sont considérées comme étant un sous-ensemble des connaissances de l'expert du domaine.

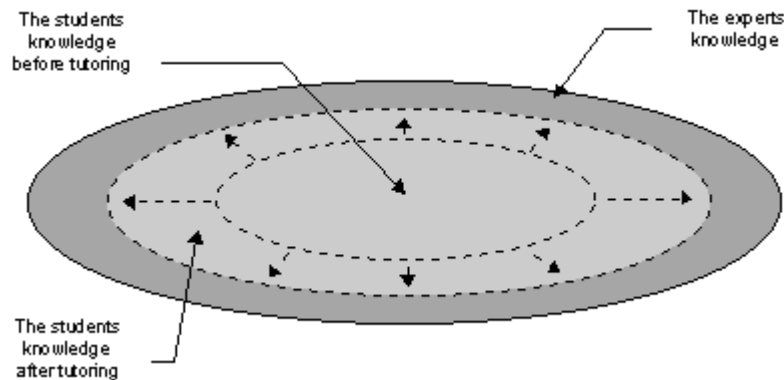


Figure 15 : La représentation d'un modèle cognitive de type "overlay" illustrant les effets du tutorat [Smith98].

Les informations au sujet de l'apprenant sont recueillies au moyen d'un réseau bayésien. Un réseau bayésien est un modèle probabiliste graphique permettant d'acquérir, de capitaliser et d'exploiter des connaissances. Il sert à **représenter la connaissance** que l'on a d'un système (technique, informatique, biologique, sociologique, économique, etc.) en vue de **prévoir** (le comportement du système), de **diagnostiquer** (les causes d'un phénomène observé dans le système), de **contrôler** (le comportement du système), de **simuler** (le comportement du système), d'**analyser** des **données** (relatives au système), de prendre des **décisions** (concernant le système), etc. Les principales raisons motivant l'utilisation des réseaux bayésiens sont leur convivialité et leur efficacité dans différents domaines.

Le système recueille des informations au sujet de l'acquisition des connaissances du domaine mais aussi au sujet des pensées réflexives de l'apprenant. C'est ce qui fait l'originalité de Prolog-Tutor.

Modèle du domaine

La conceptrice est une experte en contenu dans le domaine de la programmation. Le modèle du domaine a donc été construit sur la base de son expérience et des références qui l'accompagnent dans sa pratique.

5 TECHNIQUES UTILISEES POUR L'EVALUATION DE PROLOG-TUTOR

Trois techniques d'évaluation ont été utilisées : l'évaluation critériée, les tests pilotes et l'évaluation de l'extérieure lesquelles nous détaillons dans les sections suivantes.

5.1 L'évaluation critériée

L'évaluation critériée du système Prolog-Tutor a été basée sur une méthode d'inspection d'utilisabilité dans le domaine d'Interaction Humain-Machine (IHM) appelée "Évaluation Heuristique" de [Nielsen&Molich90] et [Pierotti94]. Selon [ISO9241-11:1998], l'utilisabilité "...is the extent to which a product can be used by specified users to achieve specified goals with *effectiveness* (i.e. accuracy and completeness with which users achieve specified goals), *efficiency* (i.e. resources expended in relation to the accuracy and completeness with which users achieve goals.), and *satisfaction* (i.e. freedom from discomfort, and positive attitudes towards the use of the product) in a specified context of use."

L'évaluation heuristique a été appliquée dans la phase de construction (prototype non terminé) correspondante aux disciplines "Implémentation" et "Test" selon le "Rational Unified Process" (RUP) [Galic&al04].

5.1.2 Description

L'évaluation heuristique permet d'inspecter une interface usager à partir d'une grille d'analyse pour détecter les aspects positifs et négatifs, du point de vue de l'utilisabilité, et ainsi de recommander des améliorations. Cette méthode d'inspection a initialement été développée par Jakob Nielsen et Rolf Molich à partir d'un ensemble de dix heuristiques basées sur des principes et recommandations issus de la recherche en ergonomie cognitive. Une heuristique est un principe général qui permet de choisir entre plusieurs choix de conception. Cette évaluation est réalisée généralement par 3 à 5 experts. Chaque expert évalue de son côté l'interface, et ensuite chacun confronte son point de vue avec celui (ceux) de son (ses) collègue (s).

L'application de ce type d'évaluation doit suivre trois stratégies : i) reconnaître le système en centralisant l'attention sur les tâches pertinentes et communes; ii) se mettre au courant des critères et recommandations ergonomiques relatif aux composantes du système; iii) effectuer un "scanning" du système et des connaissances ergonomiques disponibles.

L'évaluation heuristique est une méthode qui peut être appliquée à tous les stades du cycle de vie du produit telles que dans la phase de conception (e.g. à partir d'une maquette papier pour valider la structure et les fonctionnalités), dans une phase plus avancé du développement (e.g. sur un prototype), c'est le cas de cette évaluation de Prolog-Tutor, et finalement elle peut être appliquée dans le cadre d'une évaluation finale sur le produit fonctionnel.

Pour l'analyse de la grille, nous avons joué les rôles de deux experts ayant les connaissances appropriées dans les domaines de l'éducation et d'IHM.

La grille d'analyse (Voir section 7 Annexes) que nous avons utilisée se réfère aux 10 principes d'utilisabilité de [Nielsen&Molich90] ainsi qu'aux 3 principes d'utilisabilité de [Pierotti94] totalisant 293 problèmes d'utilisabilité ("review check-list"). Nous décrivons brièvement ces 13 principes d'utilisabilité ci-dessous:

1. *Visibilité de statut du système* : Le système devrait garder l'utilisateur toujours informé sur sa localisation par l'intermédiaire de marquages adéquats dans un temps logique. Les utilisateurs doivent savoir où ils sont et où ils doivent aller.
2. *Correspondance entre le système et le monde réel* : Le système devrait "parler" la langue des utilisateurs de façon à ce que les informations soient exposées dans un ordre naturel et logique, en s'ajustant aux conventions courantes.
3. *Contrôle de l'utilisateur et liberté*: Le système devrait obéir aux actions de l'utilisateur et fournir des "sorties de secours" explicitement marquées pour les opérations exécutées mais non souhaitées, ainsi que des options de faire/défaire les opérations dans le dialogue. Il faut aussi permettre à l'utilisateur de personnaliser sa zone de travail en lui permettant d'ajuster, par exemple, la couleur, la police de caractères, la largeur de l'écran, etc.
4. *Consistance et normalisation*: Suivre les conventions de plate-forme. Les utilisateurs ne doivent pas se questionner au sujet des significations des différents mots, situations ou actions qui veulent dire la même chose. Les titres et entêtes doivent bien représenter le contenu.
5. *Prévention des erreurs*: Le système devrait prévoir les erreurs que l'utilisateur peut faire. Il doit fournir de bons messages d'erreurs et des instructions pour les corriger faciles à comprendre afin d'éviter que des problèmes manifestent.
6. *Reconnaissance plutôt que rappel*: Le système devrait rendre les objets, les actions et les options visibles. L'utilisateur ne doit pas devoir se souvenir des informations d'une partie du dialogue à l'autre. Les directives pour l'usage du système doivent être visibles et facilement récupérables pendant le dialogue (e.g. boutons, cartes de navigation, bonnes étiquettes et liens descriptifs, etc.).
7. *Flexibilité et efficience d'utilisation*: Le système devrait fournir des raccourcis pour les utilisateurs experts et des accélérateurs afin de permettre aux utilisateurs de faire et/ou récupérer les actions plus fréquemment utilisées (e.g. des signets).
8. *"Design" esthétique et minimaliste*: Les dialogues devraient comprendre seulement les informations pertinentes et essentielles qui devraient être bien visibles. Les informations non pertinentes ou peu utilisées doivent être insérées dans les niveaux les plus bas de la hiérarchie de distribution du contenu. En fait, chaque unité d'information additionnelle dans un dialogue peut rentrer en compétition avec les unités pertinentes et réduit leur visibilité.
9. *Fonctions de l'aide (reconnaissance, diagnostic et récupération d'erreurs)* : Les messages d'erreurs devraient être exprimés dans la langue ordinaire, sans code, en indiquant précisément le problème de manière constructive, claire et compréhensible et de manière à offrir une solution aux problèmes ou en tout cas une piste pour les résoudre.

10. *Aide et documentation*: Le système devrait fournir des aides et de la documentation. Toutes les informations devraient être faciles à chercher et permettre ainsi à l'utilisateur de se concentrer sur la tâche (par exemple, tutoriels, spécifications, agents intelligents, etc.)
11. *Compétences* : Le système doit supporter, étendre, compléter, mettre en valeur les compétences, les connaissances acquises et l'expertise de l'utilisateur.
12. *Interaction Utilisateur-Système (agréable et respectueuse)* : Les interactions Utilisateur-Système doivent améliorer la qualité de vie au travail. L'utilisateur doit être traité avec respect. Le "design" doit être esthétiquement agréable à la fois dans le sens artistique et fonctionnel.
13. *Vie privée* : Le système doit aider l'utilisateur à protéger ses renseignements personnels.

Chacun des critères d'évaluation utilisés dans la grille d'analyse ont été classés sur une échelle de trois valeurs :

- YES : L'expert coche la case "YES" si le système respecte l'heuristique, c'est-à-dire, il n'y a pas de problème d'utilisabilité (majeur ou mineur).
- NO : L'expert coche la case "NO" si le système ne respecte pas l'heuristique, c'est-à-dire, il y a pas un problème d'utilisabilité.
- N/A (not applicable): L'expert coche la case "N/A" si le système ne s'applique pas à l'heuristique.

Dans le champ "Comments", nous présentons nos commentaires et nos recommandations comme on peut le voir dans l'extrait de la grille complétée présenté dans le Tableau 1.

Tableau 1 : Un extrait de la grille d'analyse (évaluation heuristique) [Pierotti94].

Review Checklist	Yes	No	N/A	Comments
1.1. Does every display begin with a title or header that describes screen contents?	●	○	○	Troisième ligne: Mettre le terme «Cours» au lieu de « Démon » afin d'indiquer clairement qu'il s'agit d'un cours complet et ainsi, associer clairement le cours et son titre «Predicate Calculus for Logic Programming». Le choix du menu «Résolution» présente à trois reprises le même titre («Résolutions») pour des parties différentes de matière. Par contre, le titre est précisé dans la fenêtre de droite dans laquelle la théorie est présentée («Résolution : Quelques notions sur les variables utilisées»). Lorsqu'on sélectionne un exercice il n'y a pas de titre qui nous indique qu'on est en train de faire un exercice. Le titre « Visualisations » n'indique pas vraiment ce qu'on peut faire. En générale, les titres du menu pourraient être plus représentatifs du contenu.
1.2. Is there a consistent icon design scheme and stylistic treatment across the system?	●	○	○	Les couleurs utilisées sont toujours les mêmes. Par exemple, le vert est toujours utilisé pour la présentation des objectifs, l'orange pour la présentation de la théorie et le rouge pour les exercices. Par contre, il aurait été plus clair d'utiliser des icônes plus représentatifs (e.g. utiliser l'icône d'un crayon).

5.1.3 Justification du choix

La méthode d'inspection d'évaluation heuristique a initialement été développée par Jakob Nielsen et Rolf Molich [Nielsen&Molich90] à partir d'un ensemble de dix heuristiques basées sur des principes et recommandations issus de la recherche en ergonomie cognitive. En effet, ils ont été les pionniers dans l'établissement des heuristiques. Néanmoins, il existe d'autres critères possibles pour l'évaluation d'interface usager qui ont été développées à partir du travail initial de [Nielsen&Molich90] telles que les Principes élémentaires de conception d'interface [Tognazzini], les Propriétés d'utilisabilité [Nigay05], les Principes de conception [Brown88] et les Critères ergonomiques [Scapin&Bastien93].

Nous avons adopté l'évaluation heuristique comme méthode d'évaluation critériée pour les raisons suivantes :

- Méthode plus utilisée et plus reconnue d'analyse de l'utilisabilité d'une interface usager dans le domaine d'IHM.
- Economique [Jeffries&al91] et facile à mettre en place : cette méthode est catégorisée comme étant de faible coût (i.e. une méthode appelée de "discount usability engineering" par [Nielsen05]) ce qui est non négligeable dans des circonstances où les ressources de budget et de temps sont limitées.

- Méthode considérée assez efficace dans l'identification des problèmes importants d'utilisabilité d'un produit.
- Aucun utilisateur n'est requis une fois que l'analyse est entamée seulement par des experts du domaine.
- Peut être réalisée individuellement.
- Nécessite une planification minimale : Cette méthode peut donc servir comme un premier diagnostic qui pourra être suivi de tests plus stricts au sujet d'un aspect spécifique de l'interface pour lequel l'évaluation heuristique aura mis à jour des problèmes.
- Utilisable très tôt dans le cycle de conception et d'implémentation du système (prototype non terminé).
- Recherches réalisées [Jeffries&al91] sur la méthode d'évaluation heuristique ont prouvé qu'il était possible d'identifier davantage de problèmes d'utilisabilité dans une interface qu'avec d'autres méthodes d'évaluation d'interface usager.
- Moins formelle que d'autres méthodes d'inspection d'interface usager.

6 RÉSULTATS DE L'ÉVALUATION

Cette section décrit l'évaluation du système selon la grille et présentation des résultats et également l'évaluation de l'extérieur (« outside assessment »).

6.1 Évaluation du système selon la grille et présentation des résultats

Les résultats de l'évaluation heuristique font état d'une part des points positifs du système à maintenir ou à renforcer et d'autre part, des points négatifs. Des commentaires sont formulés et des améliorations sont proposées.

Dans le but de présenter les résultats finaux de l'évaluation, nous avons produit une représentation formalisée des données recueillies par l'intermédiaire des graphiques en toile d'araignée (graphique en radar) présentés aux Figures 16, 17, 18, et 19. Dans la Figure 16, nous remarquons que les heuristiques « Visibilité de statut du système » (Tableau 2), « Consistance et normalisation » (Tableau 3), « Contrôle d'utilisateur » (Tableau 4) et « Aide et documentation » (Tableau 5) sont effectivement les points forts du système. D'autant que plus l'heuristique se dirige vers les extrémités supérieures de chacune des axes du graphique plus l'heuristique représente un point fort du système. Tandis que plus l'heuristique se dirige vers les extrémités inférieures des axes alors plus l'heuristique représente un point faible ou non applicable du système.

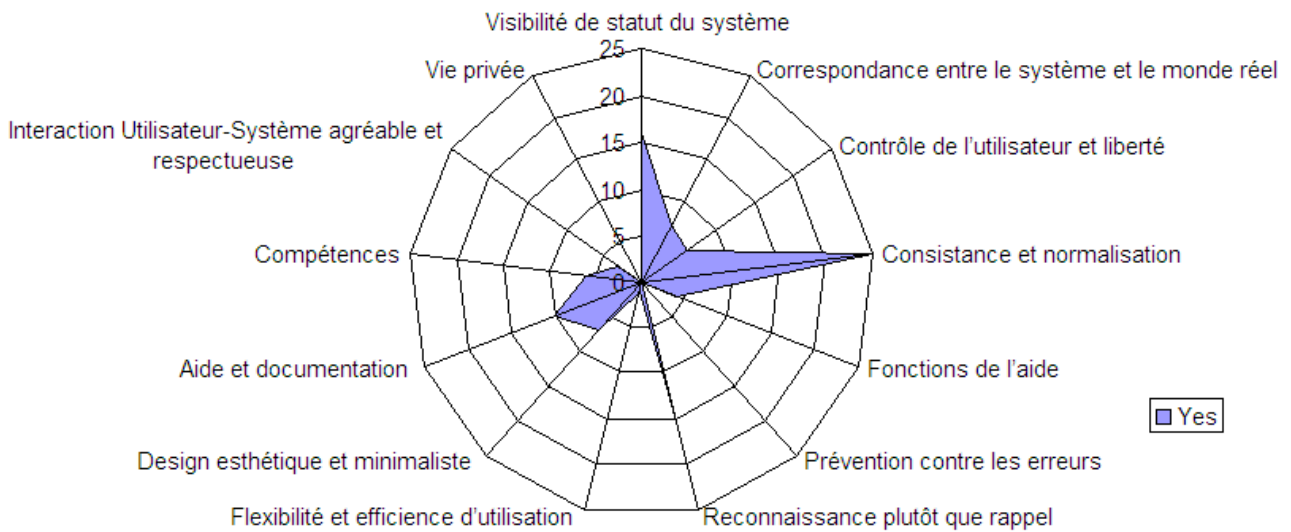


Figure 16: Les points forts du système en relation aux heuristiques.

Tableau 2 : Les principaux points forts relatifs à la "Visibilité de statut du système".

Heuristique	Points forts
1.2 Is there a consistent icon design scheme and stylistic treatment across the system?	Les couleurs utilisées sont toujours les mêmes. Par exemple, le vert est toujours utilisé pour la présentation des objectifs, l'orange pour la présentation de la théorie et le rouge pour les exercices.
1.4 Do menu instructions, prompts, and error messages appear in the same place(s) on each menu?	Le menu principal apparaît toujours au même endroit.
1.8 Is there some form of system feedback for every operator action?	Pour toutes les parties du prototype développées jusqu'à maintenant.

Tableau 3 : Les principaux points forts relatifs à la "Consistance et normalisation".

<i>Heuristique</i>	<i>Points forts</i>
4.5 Are icons labelled?	Il y a trois icônes « Outils », « Modèle Détaillé », et « Modèle Abrégé » qui sont étiquetées. Les autres boutons sont de boutons « texte ».
4.22 Are attention-getting techniques used with care? 4.23 Intensity: two levels only.	Différents niveaux d'intensité sont utilisés d'un menu à l'autre. Par exemple, dans le menu principal c'est un carré bleu qui est utilisé pour attirer l'attention. Pour les icônes on utilise une luminosité bleutée. Dans le « Reponses/Questions. Usager », on va aussi du gris au bleuté. Dans la fenêtre « Formuler un Résultat », les boutons inactivés sont déjà bleutés.

Tableau 4 : Les principaux points forts relatifs au "Contrôle d'utilisateur".

<i>Heuristique</i>	<i>Points forts</i>
3.15 If the system has multiple menu levels, is there a mechanism that allows users to go back to previous menus?	L'arborescence dans le menu principal est accessible.

Dans la Figure 17, les heuristiques « Consistance et normalisation » (Tableau 5), "Reconnaissance plutôt que rappel" (Tableau 6) et "Contrôle de l'utilisateur et liberté" (Tableau 7) représentent effectivement les points faibles du système.

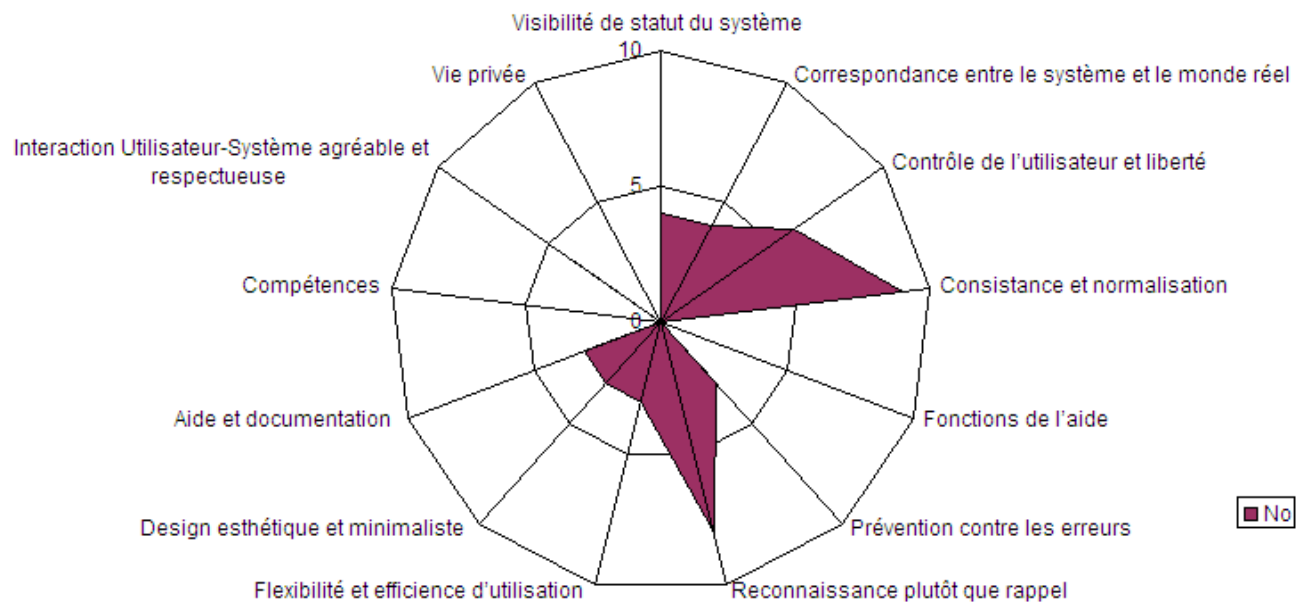


Figure 17: Les points faibles du système en relation aux heuristiques.

Tableau 5 : Les principaux points faibles relatifs à la "Consistance et normalisation".

<i>Heuristique</i>	<i>Points faibles</i>
4.16 Do embedded field-level prompts appear to the right of the field label?	La fenêtre d'explication apparaît par-dessus la fenêtre de l'énoncé.
4.22 Are attention-getting techniques used with care? 4.23 Intensity: two levels only	Différents niveaux d'intensité sont utilisés d'un menu à l'autre. Par exemple, dans le menu principal c'est un carré bleu qui est utilisé pour attirer l'attention. Pour les icônes on utilise une luminosité bleutée. Dans le "Reponses/Questions Usager", on va aussi du gris au bleuté. Dans la fenêtre «Formuler un Résultat », les boutons inactivés sont déjà bleutés.
4.29 Are attention-getting techniques used only for exceptional conditions or for time-dependent information?	À notre avis, il faudrait réduire l'utilisation de techniques pour attirer l'attention et cibler des moments privilégiés pour leur utilisation.
4.39 Are menu choice names consistent, both within each menu and across the system, in grammatical style and terminology?	Le choix du menu «Résolution » présente à trois reprises le même titre («Résolutions ») pour des parties de matières différentes.
4.51 Are high-value, high-chroma colors used to attract attention?	L'utilisation du rouge peut porter à confusion.

Tableau 6 : Les principaux points faibles relatifs à la "Reconnaissance plutôt que rappel".

<i>Heuristique</i>	<i>Points faibles</i>
7.15 Are field labels close to fields, but separated by at least one space?	Par exemple, dans la fenêtre « Une réponse » le champ pour entrer les données est loin de l'étiquette « Réponse »
7.27 Have light, bright, saturated colors been used to emphasize data and have darker, duller, and desaturated colors been used to de-emphasize data?	Dans les fenêtres « Réponses/Questions. Usager » et « Outils » les boutons ne se distinguent pas du fond. Dans la fenêtre « Feedback Cas Particulier /Domaine » le contraste entre la couleur du texte et la couleur du fond pourrait être mieux.
7.1 For question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input?	Interventions de l'apprenant plus difficile à distinguer de celles du tuteur (visuellement). Au début du « Dialogue Système/Usager » apparaît une partie « consignes » qui devraient être présentées différemment du dialogue. Par exemple, utiliser une couleur de fond différente de celle du dialogue. On doit parfois chercher dans quelle fenêtre l'information est apparue. Aérer la présentation des consignes spécialement en évitant de coller les différents boutons. Les réponses de l'apprenant se confondent avec les questions du tuteur par rapport au type de caractère utilisé.

Tableau 7 : Les principaux points faibles relatifs au "Contrôle de l'utilisateur et liberté".

<i>Heuristique</i>	<i>Points faibles</i>
3.3 In systems that use overlapping windows, is it easy for users to switch between windows?	On doit toujours déplacer les fenêtres pour avoir toutes les informations nécessaires pour compléter les exercices.
3.7 Is there an "undo" function at the level of a single action, a data entry, and a complete group of actions?	Une fois qu'on a cliqué sur le crochet turquoise, il n'est pas possible de revenir en arrière.
3.17 Can users move forward and backward between fields or dialog box options?	Il faut toujours fermer une fenêtre pour pouvoir passer à une autre fenêtre.

Il est important de souligner que certaines heuristiques se sont vues attribuer un fort pourcentage de « Not Applicable » (Figure 18). Le fait que Tutor-Prolog soit un prototype explique que certains critères soient applicables. Par exemple les fonctions d'aide restent à développer, la consistance et la normalisation sont à travailler, etc.

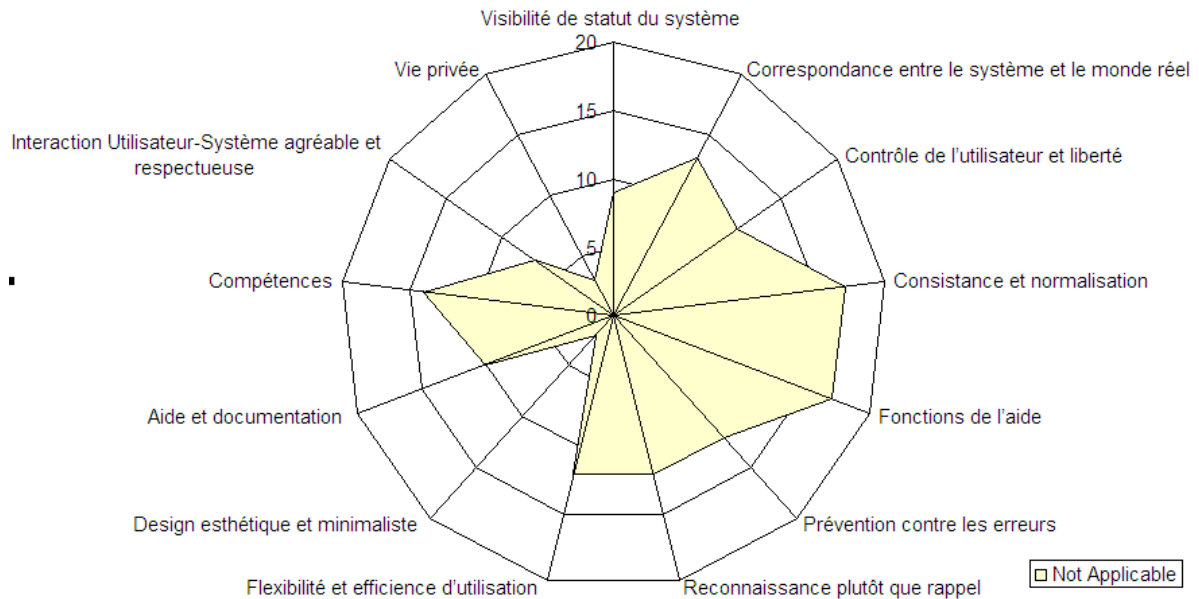


Figure 18: Les heuristiques qui ne sont pas applicables (« Not Applicable ») dans le système.

Dans la Figure 19, les trois critères évalués (i.e. Yes, No, Not Applicable) sont représentées afin de démontrer les données superposés en relation aux heuristiques.

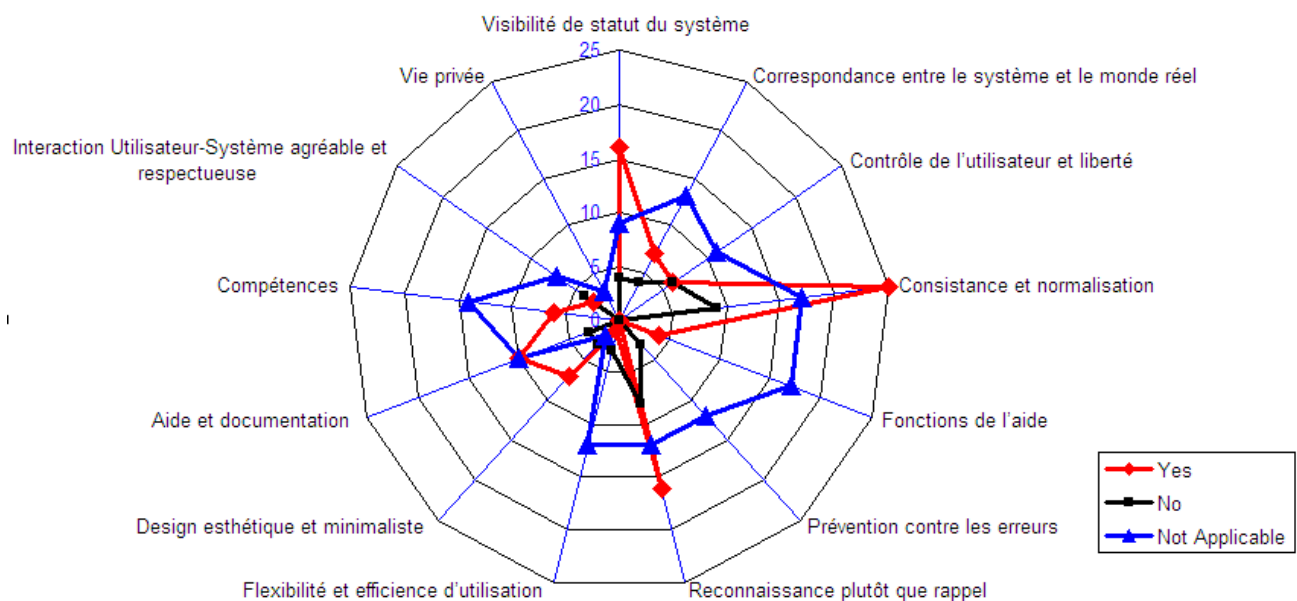


Figure 19: Les critères évalués superposés.

Dans le Tableau 8 nous présentons les divers types de problèmes d'utilisabilité qui ont été identifiés ainsi que le total de 293 problèmes d'utilisabilité collectés dans le système Prolog-Tutor.

Tableau 8 : Les résultats de l'évaluation heuristique.

<i>Heuristique appliquée</i>	<i>Nbr. de questions</i>	<i>Yes</i>	<i>No</i>	<i>Not Applicable</i>
Visibilité de statut du système	29	16	4	9
Correspondance entre le système et le monde réel	24	7	4	13
Contrôle de l'utilisateur et liberté	23	6	6	11
Consistance et normalisation	51	25	9	17
Fonctions de l'aide	21	4	0	17
Prévention contre les erreurs	15	0	3	12
Reconnaissance plutôt que rappel	41	16	8	17
Flexibilité et efficacité d'utilisation	16	1	3	12
Design esthétique et minimaliste	12	7	3	2
Aide et documentation	23	11	3	9
Compétences	21	6	1	14
Interaction Utilisateur-Système agréable et respectueuse	14	3	4	7
Vie privée	3	0	0	3
Total	293	102	48	143
	100%	35%	17%	48%

6.1.1 Discussion des résultats de l'évaluation critériée

Des trois critères utilisés (*Yes*, *No*, *Not Applicable*) les données signalent que *Not Applicable* comprend le plus haut pourcentage (48%) du système, ensuite les points forts (35%) et en dernier les points faibles (17%). Nous considérons que le haut pourcentage de 48% peut s'expliquer par le prototype qui n'est pas encore terminé et que la priorité a été accordée aux fonctionnalités du système, plus particulièrement à celles qui sont liées au diagnostic cognitif.

D'une manière générale, les points forts du système sont dans l'ordre d'importance les heuristiques *Consistance et normalisation*, *Visibilité de statut du système* et *Reconnaissance plutôt que rappel* ; les points faibles représentent *Consistance et normalisation*, *Reconnaissance plutôt que rappel* et *Contrôle de l'utilisateur et liberté*.

Les résultats les plus parlants (Figure 19 et Tableau 8) sont commentés par heuristique ci-dessous :

- *Visibilité de statut du système* : cette heuristique indique quasiment la même proportion de points forts et de *Not Applicable* probablement à cause que ce STI soit un prototype non-terminé et ainsi la priorité a été accordée aux fonctionnalités du système. Donc, dans les fonctionnalités développées, on tient compte de l'importance de la visibilité et nous croyons qu'on en tiendra compte aussi dans les futurs développements du système.
- *Contrôle de l'utilisateur et liberté* : cette heuristique présente la même proportion de points forts et de points faibles. Nous croyons qu'étant donné que Prolog-Tutor est un système tutoriel intelligent, cette heuristique peut se rendre volontairement plus stricte et limitée. Le degré de contrôle laissé à l'utilisateur dans un STI est effectivement une question de choix

pédagogique. Le but ici est d'atteindre les objectifs de haut niveau concernant l'apprentissage de l'apprenant en relation au système. Un exemple pertinent est présenté par [Nogry02] : « dans des Environnements informatiques pour l'apprentissage humain (EIAH) ayant pour objectif à court terme la résolution de problèmes, on peut privilégier un parcours tortueux du point de vue de la résolution de problèmes, plus bénéfique pour l'apprentissage, même s'il semble moins efficace qu'un parcours linéaire ».

- *Consistance et normalisation* : celle-ci représente effectivement le point fort le plus représentatif du système avec une petite proportion de points faibles. Dans un système qui vise l'acquisition de connaissances, il est très important de permettre à l'apprenant d'évoluer dans un environnement en lui permettant de se concentrer sur ses apprentissages.
- *Reconnaissance plutôt que rappel* : cette heuristique a présenté une proportion significative de points forts mais presque autant de *Not Applicable*. Donc, nous croyons que certaines fonctionnalités pourraient être ajoutées au système pour faciliter la reconnaissance.
- *Design esthétique et minimaliste* : celle-ci présente peu de points forts. Probablement en raison du fait que ce STI est un prototype non-terminé et que la priorité a été accordée aux fonctionnalités du système et non aux aspects esthétiques de l'interface.
- *Interaction Utilisateur-Système agréable et respectueuse* : cette heuristique a présenté un nombre limité de points forts et c'est probablement lié au fait qu'il faut souvent chercher l'information dans le texte et que les textes des interventions du tuteur ne sont pas tous développés.
- *Compétences* : il y a une proportion significative de *Not Applicable* probablement en raison du fait que ce STI est destiné à une clientèle particulière (les étudiants de premier cycle en Informatique) et que le système ne peut donc pas par exemple offrir, comme le suggère la grille, des fonctionnalités à la fois pour des utilisateurs experts et pour des novices.
- *Vie privée* : celle-ci a présenté *Not Applicable* dans sa totalité. Comme nous l'avons déjà mentionné, probablement que certaines fonctionnalités pourraient être ajoutées au système en relation à cette heuristique, dans une éventuelle implémentation. La protection des renseignements personnels est un facteur important concernant les risques liés aux usages des réseaux numériques et les STIs n'échappent pas à la règle. Les renseignements personnels sont ceux qui portent sur une personne physique et permettent de l'identifier. Les lois québécoises et canadiennes protègent les renseignements personnels de l'utilisateur.

6.2 Évaluation de l'extérieur (« outside assessment »)

L'évaluation de l'extérieur sert à aller consulter des experts pour vérifier des aspects précis du système. La conceptrice a eu recours à ce type d'évaluation dans le but de vérifier une hypothèse de recherche.

Hypothèse de recherche : Le contexte d'apprentissage définit la nature des connaissances qui seront diagnostiquées, la nature du processus d'apprentissage qui est observé et les principes d'instruction et de design de l'instruction qui supportent cet apprentissage. Les objectifs du DC doivent s'arimer de manière cohérente à ces éléments du contexte d'apprentissage.

Aussi, un des objectifs de la recherche de la chercheure étant de combler le fossé qui existe entre les concepteurs et les programmeurs, elle a interrogé les concepteurs pédagogiques (en tant qu'experts) à ce sujet.

L'outil CD-SPECIES

CD-SPECIES est un outil de spécification du DC qui est censé faciliter la communication entre un concepteur et un programmeur et ainsi améliorer la qualité pédagogique du système avec lequel interagit l'apprenant. Trois fonctionnalités devraient permettre de combler ce fossé de communication :

- 1) Informer le programmeur et le concepteur
- 2) Fournir un cadre de spécification au concepteur pédagogique
- 3) Assurer la cohérence des éléments d'une spécification

L'entrevue semi-structurée a été utilisée pour recueillir des informations à ces sujets. La chercheure a questionné les experts sur trois sujets principaux :

- 1) L'exhaustivité du contenu du Cadre Spécifique de Diagnostic Cognitif (CSDC)
- 2) la pertinence et l'importance du CSDC pour la conception pédagogique des EABC
- 3) la justesse et l'importance des relations entre le Diagnostic Cognitif, la pensée réflexive et les paradigmes de la cognition

Les grandes lignes d'une de ces entrevues sont décrites dans la partie intitulée « résultats ».

6.2.1 L'exhaustivité du contenu du Cadre Spécifique de Diagnostic Cognitif (CSDC)

Lors d'une formation, différents objectifs pédagogiques, de différents niveaux de difficultés, peuvent être travaillés. Prenons par exemple la taxonomie de Bloom, Benjamin S. présentée dans le Tableau 9.

Tableau 9 : Niveaux de la taxonomie de Bloom tirés de [Prégent90].

<i>Acquisition de connaissance</i>	<i>Compréhension</i>	<i>Application</i>	<i>Analyse</i>	<i>Synthèse</i>	<i>Évaluation</i>
énumérer	expliquer	résoudre	analyser	concevoir	évaluer
nommer	dire en ses propres mots	dimensionner	organiser	soutenir	juger
identifier	interpréter	utiliser	déduire	schématiser	défendre
indiquer	prévoir	manipuler	choisir	écrire	critiquer
définir	décrire	appliquer		exposer	justifier
reconnaître	comparer	calculer		discuter	
rappeler	différencier	formuler		planifier	
	Démontrer	classer			
	Prédire	modifier			
		mettre en pratique			

Le contexte peut varier en fonction du niveau de difficulté des objectifs à atteindre. Certains objectifs, de plus bas niveaux, peuvent s'enseigner au moyen d'exercices basés sur le modèle béhavioral, qui sont d'avantage du style stimulus-réponse, et cette façon de faire peut s'avérer efficace et satisfaisante pour l'apprenant. Par contre des objectifs qui tels que « défendre » peuvent se travailler au fil d'activités présentées dans différents contextes (Béhavioriste, cognitiviste et socioconstructivistes). L'atteinte d'un même objectif peut nécessiter qu'on passe d'un contexte à l'autre.

Les besoins des différents apprenants qui forment un groupe peuvent également varier en fonction de leur culture, de leurs expériences pédagogiques précédentes. Un apprenant par exemple peut avoir besoin de rétroactions courtes sur une base régulière alors qu'un autre peut avoir besoin qu'on le questionne pour stimuler sa réflexion. Certains apprenants apprennent mieux en communauté et d'autres sont dérangés par les autres. À notre avis, il est important de permettre que le diagnostic cognitif puisse s'adapter à ces variations.

Il pourrait s'avérer intéressant d'aller chercher des informations au sujet de l'apprenant en début de formation afin de « nourrir » le modèle apprenant dès le début de la formation et pouvoir ainsi adapter une partie du contenu présenté ou de pouvoir adapter certaines interventions du tuteur.

6.2.2 La pertinence et l'importance du CSDC pour la conception pédagogique des EABC

Les concepteurs pédagogiques et les programmeurs travaillent ensemble mais parlent trop souvent deux langages différents. De ce fait, les concepteurs ont parfois l'impression de ne pas être compris des programmeurs et les programmeurs ont l'impression que les concepteurs ne comprennent rien à la programmation et qu'ils ont des demandes qui ne suivent pas l'évolution de la technologie. À notre avis, le cadre de spécification pourrait effectivement servir de pont entre ces deux mondes. Il peut permettre au concepteur de comprendre les implications de ses demandes sur la tâche du programmeur et il peut permettre au programmeur de comprendre les objectifs pédagogiques poursuivis par le concepteur. Il est plus motivant de travailler lorsque nous comprenons et cherchons à atteindre les mêmes buts. Un outil comme CD-SPECIES pourrait, à notre avis, faciliter le travail des deux et ainsi améliorer la qualité du système avec lequel interragira l'apprenant.

Il est très intéressant d'établir un lien explicite entre les connaissances du domaine et le modèle de l'apprenant par le biais des questions du tuteur et des réponses de l'apprenant. Mais ça implique un travail considérable de la part du concepteur qui doit formuler les questions du dialogue tutoriel de façon claire, nette et très précise. Et ce, en plus de devoir formuler des questions en fonction des différents contextes possibles et des différents types d'apprenants. La charge de travail n'est pas moindre pour le programmeur qui doit établir le lien explicite entre toutes ces questions, le modèle des connaissances du domaine et le modèle apprenant. Il serait peut-être intéressant de laisser plus de responsabilités à l'apprenant dans l'établissement de ce diagnostic en incluant par exemple, des informations au sujet du degré d'atteinte des objectifs dans les rétroactions et en lui demandant de cocher une liste d'objectifs à travailler.

Le dialogue tutoriel semble creuser beaucoup avec l'apprenant et le pousser à réfléchir à ses stratégies aux raisons qui l'ont poussées à répondre de telle ou telle façon à une question. Cette façon de faire est très pertinente dans le cas de Prolog-Tutor car on veut que l'apprenant pense de façon logique, qu'il suive les démarches réflexives précises et nécessaires à la résolution de problèmes en programmation logique. Par contre, dans une perspective constructiviste où ne peut prévoir de quelle façon l'apprenant apprendra, il peut s'avérer inutile de le diriger ainsi dans sa façon de penser.

6.2.3 La justesse et l'importance des relations entre le Diagnostic Cognitif, la pensée réflexive et les paradigmes de la cognition

La relation entre le diagnostic cognitif et la pensée réflexive est primordiale. Elle permet à l'apprenant de réfléchir sur ses actions et d'approfondir ses connaissances. Par contre, nous croyons que leur relation avec les paradigmes de la cognition devraient être plus souples, plus ouvertes à la diversité. Comme nous l'avons mentionné plus tôt, nous ne croyons pas qu'une formation puisse être behavioriste, cognitiviste ou socioconstructiviste du début à la fin.

5 CONCLUSION

En conclusion, il est vrai que l'évaluation de SABC est un processus complexe et qu'il peut laisser des évaluateurs novices (même peut-être des experts !) sans repères solides. Il importe de développer des stratégies d'évaluation et d'adapter ou de développer des instruments fiables et sensibles aux spécificités des SABC. L'approche par l'Heuristique expérimentée dans ce travail pourrait représenter une bonne base mais devrait être adaptée à la réalité des SABC. Il faut également garder en tête que des systèmes d'une telle complexité ne peut s'évaluer au moyen d'une seule méthode et qu'il importe d'utiliser une variété de méthodes pour en évaluer tous les aspects.

L'idée du diagnostic cognitif est très intéressante puisqu'elle permet de recueillir non seulement des informations au sujet des connaissances du domaine que possède l'apprenant mais aussi au sujet de sa pensée réflexive. Le modèle apprenant ne peut qu'en être enrichi et la qualité de l'interaction améliorée.

L'évaluation critériée a fait ressortir des points forts et des points faibles de l'utilisabilité et la grille d'évaluation complétée pourrait être utilisée pour élaborer une liste des tâches à effectuer pour un programmeur qui voudrait développer une nouvelle version de Prolog-Tutor et ainsi améliorer l'utilisabilité des interfaces-usager.

Il est fortement recommandable de faire des Tests pilots pour confirmer ou infirmer ces résultats mais aussi pour voir si Prologue-Tutor peut effectivement être utilisé par des apprenants représentatifs de la clientèle cible et si ces apprenants apprennent effectivement quelque chose. À notre avis, ce système pourrait être utile dans un cours de mise à niveau en informatique au Doctorat en Informatique cognitive à l'Université du Québec à Montréal, le DIC-8000. Dans ce cours, on tente d'enseigner le langage de programmation Prologue mais on ne prévoit pas de périodes de laboratoire au cours desquelles les étudiants peuvent compléter des exercices et poser des questions à un assistant de laboratoire. Prolog-Tutor pourrait répondre au besoin d'assistance de ces étudiants.

6 RÉFÉRENCES

- 1 [Afnor96] L'Association française de Normalisation (AFNOR) (1996) X50-751-3.
< <http://www.afnor.fr/portail.asp>>
- 2 [Aïmeur&al95] Aïmeur, E. Frasson, C. & Alexe, C. (1995) "Towards New Learning Strategies in Intelligent Tutoring Systems". *Proceedings of 12th Brazilian Symposium on Artificial Intelligence, SBIA '95 Campinas, Brazil (October 11-13)*, Springer Verlag.
- 3 [Anderson83] Anderson J.R. (1983) "The Architecture of Cognition", *Harvard University Press*, Cambridge, Massachusetts (USA).
- 4 [Bellafkih&al01] Bellafkih, M., Kaitouni, B., Elhebil, F. & Ramdani, M. (2001) "Gestion neuro-flou du trafic dans un réseau local", Journée nationale : Gestion d'Imprécision et d'Incertitude dans les Systèmes à Base de Connaissances, GISBC, Mohammedia, 2001.
- 5 [Boehm81] Boehm, B. (1981) "Software Engineering Economics", Prentice-Hall.
- 6 [Boury&Tourigny97] Boury-Brisset, A. & Tourigny N. (1997) "Système d'aide à l'étude de carrefours dans le cadre des analyses de sécurité routière au ministère des Transports du Québec : Étude des solutions et recommandations", Rapport n° 3, Département d'informatique de l'Université Laval et Ministère des Transports du Québec, Québec (Canada).
- 7 [Braz04] Braz, C. (2004) "A Mcgraw Hill's Study To Go Mobile Learning Application: A Heuristic Evaluation", COMP6751 Human Computer Interface Design course, Prof. Dr. Ahmed Seffah, Concordia University, Montréal, Québec (Canada).
- 8 [Brown88] C. Marlin 'Lin' Brown (1988) "Human-Computer Interface Design Guidelines", Ablex Publishing Corporation, Norwood, NJ (USA). Lien récupéré le 10 septembre 2005
<<http://www-sop.inria.fr/acacia/ESSI/Old-Site/Principes.html>>
- 9 [Brown&Burton74] Brown, J.S. & Burton, R.R. (1978) "SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting", IBBN Report # 2790.
- 10 [Brown&Burton78] Brown, J.S. & Burton, R.R. (1978) "Diagnostic Models for Procedural Bugs in Basis Mathematical Skills", *Cognitive Science*, 2, 155-191.
- 11 [Brown&Burton79] Burton, R. R., & Brown, J. S. (1979) "An investigation of computer coaching for in formal learning activities". *International Journal of Man-Machine Studies*, 11, 5-24.
- 12 [Carr&Goldstein77] Carr, B. & Goldstein, I. (1977) "Overlays: A Theory of Modelling for Computer-Aided Instruction". *International Journal of Man-Machine Studies*, vol. 5, pp.215-236.

- 13 [Clancey82] Clancey, W. (1982) Overview of GUIDON. *Journal of Computer-Based Instruction*, 10(1/2):8-15. Summer 1983, Also in *The Handbook of Artificial Intelligence*, Volume 2, Barr and Feigenbaum (eds), Kaufmann, Los Altos (USA).
- 14 [Class&Schneider04] Class, B. ¹ & Schneider, D. ¹ () Tutorat, socio-constructivisme et capitalisation des connaissances dans un portail communautaire utilisé en éducation à distance. *Colloque sur L'industrialisation du tutorat en formation à distance*", École d'Ingénierie de la Formation à distance du CNED et la revue Distances & Savoirs en collaboration avec l'Ecole Supérieure de l'Education Nationale. ¹TECFA - TEChnologies de Formation et Apprentissage (France).
- 15 [Dewey33] Dewey, J. (1933) "How We Think. A restatement of the relation of reflective thinking to the educative process", édition révisée, Boston : D.C. Heath (USA).
- 16 [Dick87] Dick, W. (1987) "Formative Evaluation : Prospect for the Futur", *Educational Technology*, October, pp.55-57.
- 17 [Dufresne01] Dufresne, A. (2001) "Modèles et outils pour définir le soutien dans les environnements hypermédias d'apprentissage", Université de Montréal, Laboratoire en informatique cognitive et environnements de formation (LICEF). *Proceedings of the Hypermédias et Apprentissage*, Grenoble (France).
- 18 [Ernst87] Ernst, C. (1987) "Aide à la décision et systèmes experts en finance" dans *Système expert*, Editions Eyrolles.
- 19 [Frasson04] Frasson, C. (2004) « Cours IFT6243 - Concepts de Bases de données avancées - Principes généraux des systèmes intelligents qui font appel aux bases de données : les agents intelligents et les systèmes tutoriels intelligents ». Université de Montréal, Québec (Canada). Lien récupéré le 15 septembre 2005 <<http://www.iro.umontreal.ca/~frasson/documents/dift6243/6243-2004-STI.pps>>
- 20 [Galic&al04] Galic, M., Macisaac, B. & Popescu, D. (2004) "Using a Single Business Pattern with the Rational Unified Process (RUP) ", Redbooks Paper, IBM Corp. Lien récupéré le 21 septembre 2005 : <<http://www.redbooks.ibm.com/redpapers/pdfs/redp3877.pdf>>
- 21 [Huirne&Dijkhuizen88] Huirne, B. & Dijkhuizen, A. (1988) *Knowledge Based Systems in Agriculture, Prospects for Application*, Francfort, 512-520.
- 22 [Hobart95] Hobart, J. (1995) "Principles of Good GUI Design", Classic System Solutions, Inc.. Lien récupéré le 13 juillet, 2005 <http://www.classicsys.com/classic_site/articles/article_10-95.html>
- 23 [Iqbal&al99] Iqbal A.1, Oppermann R.2, Patel A. & Kinshuk, A.3 (1999) "Classification of Evaluation Methods for Intelligent Tutoring Systems", *Software Ergonomie '99 Design von Informationswelten* Pitschke), 169-181, 1University College London (UK), 2GMD-FIT (Germany, 3De Montfort University (UK), Stuttgart, Leipzig (Germany).

- 24 [ISO9241-11:1998] International Organization for Standardization (ISO) (1998) "Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs), Part 11: Guidance On Usability", Reference number ISO 9241-11:1998(E).
- 25 [Jeffries&al91] Jeffries, R., Miller, J.R., Wharton, C., & Uyeda, K.M. "User Interface Evaluation in the Real World: A Comparison of Four Techniques". Proc. ACM CHZ'91 (New Orleans, LA, 27 April-2 May 1991), 119-124.
- 26 [Lurain94] Lurain, M. U. (1994) "Intelligent Tutoring Systems: An Historic Review in the Context of the Development of Artificial Intelligence and Educational Psychology ", Division of Science and Mathematics Education, College of Natural Science, Michigan State University, MI (USA).
- 27 [Mark &Greer92] Mark, M. & Greer, J. (1992) Evaluation Methodologies for Intelligent Tutorial Systems, ARIES Laboratory, Department of Computational Science, University of Saskatchewan, Saskatoon (Canada).
- 28 [Marlet05] Marlet, R. (2005) "Génie logicielle - Cycle de vie", LaBRI/INRIA. Lien récupéré le 20 septembre 2005
<<http://www.labri.fr/perso/marlet/teaching/GenLog/handouts/GenLogCycleVie-4p.pdf>>
- 29 [Marton&Harvey94] Marton, Ph. & Harvey, D. (1994) "L'évaluation des systèmes d'apprentissage multimédia interactif », *Educatechnologies*, no 3, pp.15-27.
- 30 [Mengelle&Frasson96] Mengelle, T. & Frasson, C. (1996) "A Multi-Agent Architecture for an ITS with Multiple Strategies", Université de Montréal, Département d'informatique et de recherche Opérationnelle, Montréal, Québec (Canada).
- 31 [Nielsen05] Nielsen, J. (2005) "useit.com" Jacob Nielsen Web site. Lien récupéré le 27 septembre 2005 : <<http://www.useit.com/>>
- 32 [Nielsen&Molich90] Nielsen, J. and Molich, R. (1990) "Heuristic Evaluation of User Interfaces". In *Proceedings of ACM CHI'90 Conference*. (Seattle, WA, 1-5 April) 249-256 (USA).
- 33 [Nigay05] C. Nigay, L. (2005) "Propriétés d'utilisabilité" (Cours), Université de Joseph Fourier, Grenoble (France).
- 34 [Nkambou05] Nkambou, R. (2005) "Notes des cours : DIC9340 Systèmes d'apprentissage à base de connaissances", Université Du Québec à Montréal, Département d'informatique, Montréal, Québec (Canada).
- 35 [Nogry&al02] Nogry, S., Jean-Daubias, S. & Ollagnier-Beldame, M. () « Évaluation des EIAH : une nécessaire diversité des méthodes », LIRIS – ERTé, Université Claude Bernard Lyon 1, CNRS Nautibus, Campus de la Doua, Villeurbanne Cedex (France).
- 36 [Perkins91] Perkins, D.N. (1991) "Technology Meets Constructivism: Do they make a marriage?", *Educational Technology*, 31(5), 18-23.

- 37 [Piaget78] Piaget, J. (1978) "Behaviour and Evolution", New York: Pantheon Books.
- 38 [Pierotti94] Pierotti, D. (1996) Xerox Corporation. "How to Conduct a Heuristic Evaluation A System Check-List", Usability Analysis & Design, Version 1.0.
- 39 [Prégent90] Prégent, R. (1990) « La préparation d'un cours » Montréal : Éditions de l'École polytechnique, Montréal, Québec (Canada).
- 40 [Retschitzki95] Retschitzki, J. (1995) "Apprendre avec l'informatique: Vers une révolution?", Institut de psychologie, Université de Fribourg (Suisse). Lien récupéré le 13 juillet, 2005 <<http://www.unifr.ch/spc/UF/95juin/retschitzki.html>>
- 41 [Savard96] Savard, I. (2002) "Évaluation formative en cours de développement » du SAMI cognitive aux niveaux de l'efficacité et de l'attitude, Montréal, Québec (Canada).
- 42 [Scapin&Bastien93] Bastien, J.M.C. & Scapin, D. (1993) "Ergonomic Criteria for the Evaluation of Human-Computer Interfaces ", Institut National de recherche en informatique et en automatique (INRIA), (France).
- 43 [Smith98] Smith, S. (1998) "Intelligent Tutoring Systems", School of Computing Science Middlesex University, London (UK). Lien récupéré le 14 juillet 2005: <<http://www.cs.mdx.ac.uk/staffpages/serengul/Traditional.Computer.Aided.Learning.Systems.htm>>
- 44 [Tchetagni&al05] Tchetagni, J.1,2, Nkambou, R.1 & Bourdeu, J.2 (2005) "Supporting Student Reflection in an Intelligent Tutoring System for Logic Programming ", 1Computer Science Department, University of Quebec, Montreal, Canada; 2 LICEF Research Center, Tele-University, Montreal (Canada).
- 45 [Tognazzini05] Tognazzini, B. (2005) "Principes élémentaires de conception d'interface". Lien récupéré le 14 juillet 2005 < <http://www.asktog.com/basics/firstPrinciples.html>>
- 46 [Wenger87] Wenger, E. (1987) "Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge", Morgan-Kaufmann Publishers Inc., Los Altos, CA (USA).
- 47 [Wikipédia05] Wikipédia – L'encyclopédie libre (2005) « Cycles de développement » . Lien récupéré le 11 août 2005: <http://fr.wikipedia.org/wiki/Cycle_de_d%C3%A9veloppement#Cycle_it.C3.A9ratif>
- 48 [Woolf86] Woolf, B. (1986) "Tutoring a Complex Industrial Process". In *Proceedings of the 4th National Conference on Artificial Intelligence (AAAI-86)*. Palo Alto, CA: Morgan Kaufmann (USA).

7 ANNEXES

HEURISTIC EVALUATION - A SYSTEM CHECKLIST

1. VISIBILITY OF SYSTEM STATUS

The system should always keep user informed about what is going on, through appropriate feedback within reasonable time.

Review Checklist	Yes	No	N/A	Comments
1.1. Does every display begin with a title or header that describes screen contents?	☒	☐	☐	Troisième ligne: Mettre le terme "Cours" au lieu de « Démo » afin d'indiquer clairement qui s'agit d'un cours complet et ainsi, associer clairement le cours et son titre «Predicate Calculus for Logic Programming». Le choix du menu «Résolution » présente à trois reprises le même titre («Résolutions ») pour des parties différentes de matière. Par contre, le titre est précisé dans la fenêtre de droite dans laquelle la théorie est présentée (« Résolution : Quelques notions sur les variables utilisées »). Lorsqu'on sélectionne un exercice il n'y a pas de titre qui nous indique qu'on est en train de faire un exercice. Le titre « Visualisations » n'indique pas vraiment ce qu'on peut faire. En générale, les titres du menu pourraient être plus représentatifs du contenu.
1.2 Is there a consistent icon design scheme and stylistic treatment across the system?	☒	☐	☐	Les couleurs utilisées sont toujours les mêmes. Par exemple, le vert est toujours utilise pour la présentation des objectifs, l'orange pour la présentation de la théorie et le rouge pour les exercices. Par contre, il aurait été plus clair d'utiliser des icônes plus représentatifs (e.g. utiliser l'icône d'un crayon).
1.3 Is a single, selected icon clearly visible when surrounded by unselected icons?	☒	☐	☐	

1.4 Do menu instructions, prompts, and error messages appear in the same place(s) on each menu?	☒ ☐ ☐	Le menu principal apparaît toujours au même endroit. Les prompts qui permettent à l'apprenant de répondre aux questions des exercices n'apparaissent pas à la bonne place puisqu'ils cachent toujours les informations de l'énoncé dont l'apprenant a besoin pour répondre. Nous n'avons pas vu apparaître de messages d'erreur.
1.5 In multi-page data entry screens, is each page labelled to show its relation to others?	☐ ☐ ☒	
1.6 If overtype and insert mode are both available, is there a visible indication of which one the user is in?	☐ ☐ ☒	
1.7 If pop-up windows are used to display error messages, do they allow the user to see the field in error?	☐ ☐ ☒	Nous n'avons pas vu apparaître de messages d'erreur.
1.8 Is there some form of system feedback for every operator action?	☒ ☐ ☐	Pour toutes les parties du prototype développées jusqu'à maintenant.
1.9 After the user completes an action (or group of actions), does the feedback indicate that the next group of actions can be started?	☒ ☐ ☐	Par contre, l'apprenant doit s'arrêter pour réfléchir aux choix qui s'offrent à lui parce que l'interface est très chargée.
1.10 Is there visual feedback in menus or dialog boxes about which choices are selectable?	☐ ☒ ☐	
1.11 Is there visual feedback in menus or dialog boxes about which choice the cursor is on now?	☒ ☐ ☐	Oui pour le menu principal. Par contre, dans l'interface des exercices, le bouton sélectionné ne se distingue pas des autres.
1.12 If multiple options can be selected in a menu or dialog box, is there visual feedback about which options are already selected?	☐ ☒ ☐	Seul le choix immédiat est mis en évidence. Les parties déjà consultées par l'apprenant ne se distinguent pas de celles qu'il n'a jamais consultées.
1.13 Is there visual feedback when objects are selected or moved?	☒ ☐ ☐	
1.14 Is the current status of an icon clearly indicated?	☒ ☐ ☐	
1.15 Is there feedback when function keys are pressed?	☐ ☐ ☒	

1.16 If there are observable delays (greater than fifteen seconds) in the system's response time, is the user kept informed of the system's progress?	☐ ☐ ☒	
1.17 Are response times appropriate to the task?	☒ ☐ ☐	
1.18 Typing, cursor motion, mouse selection: 50-150 milliseconds	☒ ☐ ☐	
1.19 Simple, frequent tasks: less than 1 second	☒ ☐ ☐	
1.20 Common tasks: 2-4 seconds	☒ ☐ ☐	
1.21 Complex tasks: 8-12 seconds	☐ ☐ ☒	
1.22 Are responses times appropriate to the users' cognitive processing?	☒ ☐ ☐	
1.23 Continuity of thinking is required and information must be remembered throughout several responses: less than two seconds.	☐ ☐ ☒	Dans le « Dialogue Système/Usager » on fait appel à une démarche pédagogique dans laquelle le rythme de l'apprenant doit être respecté. Nous considérons inutile d'évaluer le temps.
1.24 High levels of concentration aren't necessary and remembering information is not required: two to fifteen seconds.	☐ ☐ ☒	
1.25 Is the menu-naming terminology consistent with the user's task domain?	☐ ☒ ☐	En générale, les titres des menus pourraient être plus représentatifs du contenu présenté. Par exemple, dans l'activité « Résolution » trois choix portent le même titre (Résolution).
1.26 Does the system provide <i>visibility</i> : that is, by looking, can the user tell the state of the system and the alternatives for action?	☒ ☐ ☐	Oui, mais ça ne se fait pas instantanément car la charge cognitive est élevée.
1.27 Do GUI menus make obvious which item has been selected?	☒ ☐ ☐	Le choix immédiat est mis en évidence. Les parties déjà consultées par l'apprenant ne se distinguent pas de celles qu'il n'a jamais consultées.
1.28 Do GUI menus make obvious whether deselection is possible?	☐ ☐ ☒	

1.29 If users must navigate between multiple screens, does the system use context labels, menu maps, and place markers as navigational aids?	☐ ☒ ☐	
--	--	--

2. MATCH BETWEEN SYSTEM AND THE REAL WORLD

The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Review Checklist	Yes	No	N/A	Comments
2.1 Are icons concrete and familiar?	☐	☒	☐	
2.2 Are menu choices ordered in the most logical way, given the user, the item names, and the task variables?	☒	☐	☐	Le contenu, les activités, les objectifs et la théorie nous semble être présenté d'une façon logique.
2.3 If there is a natural sequence to menu choices, has it been used?	☒	☐	☐	
2.4 Do related and interdependent fields appear on the same screen?	☐	☐	☒	
2.5 If shape is used as a visual cue, does it match cultural conventions?	☐	☐	☒	
2.6 Do the selected colors correspond to common expectations about color codes?	☐	☒	☐	Utiliser des couleurs claires (bouton actif) et foncées (bouton déjà sélectionnés). Dans le menu principal, on utilise le rouge pour représenter les exercices en général. Cette couleur est généralement utilisée pour indiquer une erreur. Des apprenants pourraient penser que ces exercices n'ont pas été réussis. Dans le « Dialogue Système/Usager » de la partie exercice, on pourrait utiliser un code de couleurs pour distinguer les rétroactions positives des négatives. Les rétroactions du tuteur sont toujours présentées en rouge. Les questions, les rétroactions et les résumés du tuteur se distinguent les uns des autres et c'est bien.

2.7 When prompts imply a necessary action, are the words in the message consistent with that action?	☐ ☒ ☐	Par exemple, dans le « Réponses/questions. Usager » il est difficile de comprendre la différence entre les boutons “Résultat=” et “Réponse=". Et, lorsqu'on clique sur le bouton “Résultat=” et que la fenêtre correspondante s'affiche on ne comprend pas ce qu'il faut faire.
2.8 Do keystroke references in prompts match actual key names?	☐ ☐ ☒	
2.9 On data entry screens, are tasks described in terminology familiar to users?	☐ ☐ ☒	Par exemple, dans la fenêtre “Formuler un Résultat” la tâche n'est pas décrite. La question (tâche) est toujours présentée dans la partie « Dialogue Système/Usager » et, dans la fenêtre où les résultats doivent être entrés, on ne nous explique pas (visuellement ou textuellement) quelle est la tâche à accomplir.
2.10 Are field-level prompts provided for data entry screens?	☐ ☐ ☒	
2.11 For question and answer interfaces, are questions stated in clear, simple language?	☒ ☐ ☐	Le langage utilisé respecte la théorie du langage de programmation Prolog.
2.12 Do menu choices fit logically into categories that have readily understood meanings?	☒ ☐ ☐	
2.13 Are menu titles parallel grammatically?	☒ ☐ ☐	Oui mais les articles présents dans certains titres pourraient être enlevés pour assurer une consistance.
2.14 Does the command language employ user jargon and avoid computer jargon?	☐ ☐ ☒	
2.15 Are command names specific rather than general?	☐ ☐ ☒	
2.16 Does the command language allow both full names and abbreviations?	☐ ☐ ☒	
2.17 Are input data codes meaningful?	☒ ☐ ☐	
2.18 Have uncommon letter sequences been avoided whenever possible?	☐ ☐ ☒	
2.19 Does the system automatically enter leading or trailing spaces to align decimal points?	☐ ☐ ☒	

2.20 Does the system automatically enter a dollar sign and decimal for monetary entries?	☐	☐	☒	
2.21 Does the system automatically enter commas in numeric values greater than 9999?	☐	☐	☒	
2.22 Do GUI menus offer activation: that is, make obvious how to say "now do it"?	☐	☒	☐	
Review Checklist	Yes	No	N/A	Comments
2.23 Has the system been designed so that keys with similar names do not perform opposite (and potentially dangerous) actions?	☒	☐	☐	
2.24 Are function keys labelled clearly and distinctively, even if this means breaking consistency rules?	☐	☐	☒	

3. USER CONTROL AND FREEDOM

Users should be free to select and sequence tasks (when appropriate), rather than having the system does this for them. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The system should support undo and redo.

Review Checklist	Yes	No	N/A	Comments
3.1 If setting up windows is a low-frequency task, is it particularly easy to remember?	☐	☐	☒	Il y a beaucoup de fenêtres à réarranger dans le système donc une tâche très fréquente.
3.2 In systems that use overlapping windows, is it easy for users to rearrange windows on the screen?	☒	☐	☐	Oui toutes les fenêtres sont faciles à manipuler à l'exception de celle qui apparaît lorsqu'on sélectionne un item du menu « Capacités visées dans Prolog-Tutor ».
3.3 In systems that use overlapping windows, is it easy for users to switch between windows?	☐	☒	☐	On doit toujours déplacer les fenêtres pour avoir toutes les informations nécessaires pour compléter les exercices.
3.4 When a user's task is complete, does the system wait for a signal from the user before processing?	☒	☐	☐	
3.5 Can users type-ahead in a system with many nested menus?	☐	☐	☒	

3.6 Are users prompted to confirm commands that have drastic, destructive consequences?	☐ ☐ ☒	
3.7 Is there an "undo" function at the level of a single action, a data entry, and a complete group of actions?	☐ ☒ ☐	Une fois qu'on a cliqué sur le crochet turquoise, il n'est pas possible de revenir en arrière.
3.8 Can users cancel out of operations in progress?	☐ ☐ ☒	
3.9 Are character edits allowed in commands?	☐ ☐ ☒	
3.10 Can users reduce data entry time by copying and modifying existing data?	☐ ☐ ☒	
3.11 Are character edits allowed in data entry fields?	☒ ☐ ☐	
3.12 If menu lists are long (more than seven items), can users select an item either by moving the cursor or by typing a mnemonic code?	☒ ☐ ☐	
3.13 If the system uses a pointing device, do users have the option of either clicking on menu items or using a keyboard shortcut?	☐ ☐ ☒	
3.14 Are menus broad (many items on a menu) rather than deep (many menu levels)?	☒ ☐ ☐	
3.15 If the system has multiple menu levels, is there a mechanism that allows users to go back to previous menus?	☒ ☐ ☐	L'arborescence dans le menu principal est accessible.
3.16 If users can go back to a previous menu, can they change their earlier menu choice?	☐ ☐ ☒	
3.17 Can users move forward and backward between fields or dialog box options?	☐ ☒ ☐	Il faut toujours fermer une fenêtre pour pouvoir passer à une autre fenêtre.
3.18 If the system has multi-page data entry screens, can users move backward and forward among all the pages in the set?	☐ ☐ ☒	
3.19 If the system uses a question and answer interface, can users go back to previous questions or skip forward to later questions?	☐ ☒ ☐	
3.20 Do function keys that can cause serious consequences have an undo feature?	☐ ☐ ☒	

3.21 Can users easily reverse their actions?	☐ ☒ ☐	Selon nous c'est un choix pédagogique.
3.22 If the system allows users to reverse their actions, is there a retracing mechanism to allow for multiple undos?	☐ ☐ ☒	
3.23 Can users set their own system, session, file, and screen defaults?	☐ ☒ ☐	

4. CONSISTENCY AND STANDARDS

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Review Checklist	Yes	No	N/A	Comments
4.1 Have industry or company formatting standards been followed consistently in all screens within a system?	☐	☒	☐	
4.2 Has a heavy use of all uppercase letters on a screen been avoided?	☒	☐	☐	
4.3 Do abbreviations not include punctuation?	☐	☐	☒	
4.4 Are integers right-justified and real numbers decimal-aligned?	☐	☐	☒	
4.5 Are icons labelled?	☒	☐	☐	Il y a trois icônes « Outils », « Modèle Détaillé », et « Modèle Abrégé » qui sont étiquetées. Les autres boutons sont de boutons textes.
4.6 Are there no more than twelve to twenty icon types?	☒	☐	☐	
4.7 Are there salient visual cues to identify the active window?	☒	☐	☐	
4.8 Does each window have a title?	☒	☐	☐	
4.9 Is vertical and horizontal scrolling possible in each window?	☐	☒	☐	

4.10 Does the menu structure match the task structure?	☒ ☐ ☐	
4.11 Have industry or company standards been established for menu design, and are they applied consistently on all menu screens in the system?	☐ ☒ ☐	
4.12 Are menu choice lists presented vertically?	☒ ☐ ☐	
4.13 If "exit" is a menu choice, does it always appear at the bottom of the list?	☐ ☐ ☒	
4.14 Are menu titles either centered or left-justified?	☒ ☐ ☐	
4.15 Are menu items left-justified, with the item number or mnemonic preceding the name?	☐ ☐ ☒	
4.16 Do embedded field-level prompts appear to the right of the field label?	☐ ☒ ☐	Parfois la fenêtre d'explication apparaît par-dessus l'étiquette.
4.17 Do on-line instructions appear in a consistent location across screens?	☐ ☐ ☒	
4.18 Are field labels and fields distinguished typographically?	☒ ☐ ☐	
4.19 Are field labels consistent from one data entry screen to another?	☒ ☐ ☐	
4.20 Are fields and labels left-justified for alpha lists and right-justified for numeric lists?	☐ ☐ ☒	
4.21 Do field labels appear to the left of single fields and above list fields?	☒ ☐ ☐	
4.22 Are attention-getting techniques used with care?	☐ ☒ ☐	
4.23 Intensity: two levels only	☐ ☒ ☐	Différents niveaux d'intensité sont utilisés d'un menu à l'autre. Par exemple, dans le menu principal c'est un carré bleu qui est utilisé pour attirer l'attention. Pour les icônes on utilise une luminosité bleutée. Dans le "Reponses/Questions. Usager", on va aussi du gris au bleuté. Dans la fenêtre «Formuler un Résultat », les boutons inactivés sont déjà bleutés.

4.24 Size: up to four sizes	☒ ☐ ☐	
4.25 Font: up to three	☒ ☐ ☐	
4.26 Blink: two to four hertz	☐ ☐ ☒	
4.27 Color: up to four (additional colors for occasional use only)	☒ ☐ ☐	
4.28 Sound: soft tones for regular positive feedback, harsh for rare critical conditions	☐ ☐ ☒	
4.29 Are attention-getting techniques used only for exceptional conditions or for time-dependent information?	☐ ☒ ☐	À notre avis, il faudrait réduire l'utilisation de techniques pour attirer l'attention et cibler des moments privilégiés pour leur utilisation.
4.30 Are there no more than four to seven colors, and are they far apart along the visible spectrum?	☒ ☐ ☐	
4.31 Is a legend provided if color codes are numerous or not obvious in meaning?	☒ ☐ ☐	Le code de couleurs utilisé dans le menu principal est important et il n'est pas facilement accessible et ne se consulte pas rapidement.
4.32 Have pairings of high-chroma, spectrally extreme colors been avoided?	☒ ☐ ☐	
4.33 Are saturated blues avoided for text or other small, thin line symbols?	☐ ☒ ☐	Dans la fenêtre « Feedback Cas Particulier Domaine » on utilise du texte bleu sur fond bleu.
4.34 Is the most important information placed at the beginning of the prompt?	☐ ☒ ☐	
4.35 Are user actions named consistently across all prompts in the system?	☐ ☒ ☐	
4.36 Are system objects named consistently across all prompts in the system?	☒ ☐ ☐	
4.37 Do field-level prompts provide more information than a restatement of the field name?	☒ ☐ ☐	
4.38 For question and answer interfaces, are the valid inputs for a question listed?	☒ ☐ ☐	Pour les questions de type « choix multiples ».

4.39 Are menu choice names consistent, both within each menu and across the system, in grammatical style and terminology?	☒ ☐ ☐	Par contre, le choix du menu «Résolution» présente à trois reprises le même titre («Résolutions») pour des parties de matières différentes.
4.40 Does the structure of menu choice names match their corresponding menu titles?	☐ ☐ ☒	
4.41 Are commands used the same way, and do they mean the same thing, in all parts of the system?	☐ ☐ ☒	
4.42 Does the command language have a consistent, natural, and mnemonic syntax?	☐ ☐ ☒	
4.43 Do abbreviations follow a simple primary rule and, if necessary, a simple secondary rule for abbreviations that otherwise would be duplicates?	☐ ☐ ☒	
4.44 Is the secondary rule used only when necessary?	☐ ☐ ☒	
4.45 Are abbreviated words all the same length?	☐ ☐ ☒	
4.46 Is the structure of a data entry value consistent from screen to screen?	☒ ☐ ☐	
4.47 Is the method for moving the cursor to the next or previous field consistent throughout the system?	☒ ☐ ☐	
4.48 If the system has multi-page data entry screens, do all pages have the same title?	☐ ☐ ☒	
4.49 If the system has multi-page data entry screens, does each page have a sequential page number?	☐ ☐ ☒	
4.50 Does the system follow industry or company standards for function key assignments?	☐ ☐ ☒	
4.51 Are high-value, high-chroma colors used to attract attention?	☒ ☐ ☐	L'utilisation du rouge peut porter à confusion.

5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS

Error messages should be expressed in plain language (NO CODES).

Review Checklist	Yes	No	N/A	Comments
5.1 Is sound used to signal an error?	☐	☐	☒	
5.2 Are prompts stated constructively, without overt or implied criticism of the user?	☒	☐	☐	Oui mais dans la partie « Dialogue Système/Usager » le tuteur nous a répondu que « La question ne peut pas être plus claire ».
5.3 Do prompts imply that the user is in control?	☒	☐	☐	
5.4 Are prompts brief and unambiguous?	☒	☐	☐	
5.5 Are error messages worded so that the system, not the user, takes the blame?	☐	☐	☒	
5.6 If humorous error messages are used, are they appropriate and inoffensive to the user population?	☐	☐	☒	
5.7 Are error messages grammatically correct?	☐	☐	☒	
5.8 Do error messages avoid the use of exclamation points?	☐	☐	☒	
5.9 Do error messages avoid the use of violent or hostile words?	☐	☐	☒	
5.10 Do error messages avoid an anthropomorphic tone?	☐	☐	☒	
5.11 Do all error messages in the system use consistent grammatical style, form, terminology, and abbreviations?	☐	☐	☒	
5.12 Do messages place users in control of the system?	☒	☐	☐	
5.13 Does the command language use normal action-object syntax?	☐	☐	☒	

5.14 Does the command language avoid arbitrary, non-English use of punctuation, except for symbols that users already know?	☐	☐	☒	
5.15 If an error is detected in a data entry field, does the system place the cursor in that field or highlight the error?	☐	☐	☒	
5.16 Do error messages inform the user of the error's severity?	☐	☐	☒	
5.17 Do error messages suggest the cause of the problem?	☐	☐	☒	
5.18 Do error messages provide appropriate semantic information?	☐	☐	☒	
5.19 Do error messages provide appropriate syntactic information?	☐	☐	☒	
5.20 Do error messages indicate what action the user needs to take to correct the error?	☐	☐	☒	
5.21 If the system supports both novice and expert users, are multiple levels of error-message detail available?	☐	☐	☒	

6. ERROR PREVENTION

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Review Checklist	Yes	No	N/A	Comments
6.1 If the database includes groups of data, can users enter more than one group on a single screen?	☐	☐	☒	
6.2 Have dots or underscores been used to indicate field length?	☐	☐	☒	
6.3 Is the menu choice name on a higher-level menu used as the menu title of the lower-level menu?	☐	☐	☒	
6.4 Are menu choices logical, distinctive, and mutually exclusive?	☐	☒	☐	

6.5 Are data inputs case-blind whenever possible?	☐	☐	☒	
6.6 If the system displays multiple windows, is navigation between windows simple and visible?	☐	☒	☐	
6.7 Are the function keys that can cause the most serious consequences in hard-to-reach positions?	☐	☐	☒	
6.8 Are the function keys that can cause the most serious consequences located far away from low-consequence and high-use keys?	☐	☐	☒	
6.9 Has the use of qualifier keys been minimized?	☐	☐	☒	
6.10 If the system uses qualifier keys, are they used consistently throughout the system?	☐	☐	☒	
6.11 Does the system prevent users from making errors whenever possible?	☐	☐	☒	
6.12 Does the system warn users if they are about to make a potentially serious error?	☐	☐	☒	
6.13 Does the system intelligently interpret variations in user commands?	☐	☐	☒	
6.14 Do data entry screens and dialog boxes indicate the number of character spaces available in a field?	☐	☒	☐	
6.15 Do fields in data entry screens and dialog boxes contain default values when appropriate?	☐	☐	☒	

7. RECOGNITION RATHER THAN RECALL

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Review Checklist	Yes	No	N/A	Comments
7.1 For question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input?	☐	☒	☐	Au début du « Dialogue Système/Usager » apparaît une partie « consignes » qui devrait être présentée différemment du dialogue. Par exemple, utiliser une couleur de fond différente de celle du dialogue. Airer la présentation des consignes spécialement en évitant de coller les différents boutons. Les réponses de l'apprenant se confondent avec les questions du tuteur par rapport au type de caractère utilisé. Cependant, des indices visuels sont utilisés pour illustrer les différents types d'intervention du tuteur. Par exemple, les questions sont en bleu et les rétroactions sont en rouge.
7.2 Does the data display start in the upper-left corner of the screen?	☒	☐	☐	
7.3 Are multiword field labels placed horizontally (not stacked vertically)?	☒	☐	☐	
7.4 Are all data a user needs on display at each step in a transaction sequence?	☐	☐	☒	
7.5 Are prompts, cues, and messages placed where the eye is likely to be looking on the screen?	☒	☐	☐	
7.6 Have prompts been formatted using white space, justification, and visual cues for easy scanning?	☒	☐	☐	Il y a place à amélioration.
7.7 Do text areas have "breathing space" around them?	☒	☐	☐	Il y a place à amélioration.
7.8 Is there an obvious visual distinction made between "choose one" menu and "choose many" menus?	☐	☐	☒	
7.9 Have spatial relationships between soft function keys (on-screen cues) and keyboard	☐	☐	☒	

function keys been preserved?			
7.10 Does the system gray out or delete labels of currently inactive soft function keys?	☐	☐	☒
7.11 Is white space used to create symmetry and lead the eye in the appropriate direction?	☐	☒	☐
7.12 Have items been grouped into logical zones, and have headings been used to distinguish between zones?	☒	☐	☐
7.13 Are zones no more than twelve to fourteen characters wide and six to seven lines high?	☐	☐	☒
7.14 Have zones been separated by spaces, lines, color, letters, bold titles, rules lines, or shaded areas?	☒	☐	☐
7.15 Are field labels close to fields, but separated by at least one space?	☐	☒	☐
7.16 Are long columnar fields broken up into groups of five, separated by a blank line?	☐	☒	☐
7.17 Are optional data entry fields clearly marked?	☐	☐	☒
7.18 Are symbols used to break long input strings into "chunks"?	☐	☐	☒
7.19 Is reverse video or color highlighting used to get the user's attention?	☒	☐	☐
7.20 Is reverse video used to indicate that an item has been selected?	☐	☒	☐
7.21 Is size, boldface, underlining, color, shading, or typography used to show relative quantity or importance of different screen items?	☒	☐	☐
7.22 Are borders used to identify meaningful groups?	☒	☐	☐
7.23 Has the same color been used to group related elements?	☒	☐	☐

Par exemple, dans la fenêtre « Une réponse » le champ pour entrer les données est loin de l'étiquette « Réponse »

Dans la fenêtre « Capacités visées dans Prolog-Tutor » toutes les informations sont présentées dans un seul paquet.

Cependant dans la fenêtre « Dialogue Système/Usager » les lignes pourraient être utilisées pour identifier visuellement les différents sujets du dialogue.

7.24 Is color coding consistent throughout the system?	☒ ☐ ☐	
7.25 Is color used in conjunction with some other redundant cue?	☒ ☐ ☐	Par exemple, dans la théorie « Résolution avec Coupure » on utilise le rouge en conjonction avec la majuscule.
7.26 Is there good color and brightness contrast between image and background colors?	☐ ☐ ☒	
7.27 Have light, bright, saturated colors been used to emphasize data and have darker, duller, and desaturated colors been used to de-emphasize data?	☐ ☒ ☐	Dans les fenêtres « Réponses/Questions. Usager » et « Outils » les boutons ne se distinguent pas du fond. Dans la fenêtre « Feedback Cas Particulier /Domaine » le contraste entre la couleur du texte et la couleur du fond pourrait être mieux.
7.28 Is the first word of each menu choice the most important?	☐ ☒ ☐	Parfois des articles sont présentés en premier.
7.29 Does the system provide <i>mapping</i> : that is, are the relationships between controls and actions apparent to the user?	☐ ☒ ☐	Nous savons que c'est prévu pour une implémentation future.
7.30 Are input data codes distinctive?	☐ ☐ ☒	
7.31 Have frequently confused data pairs been eliminated whenever possible?	☐ ☐ ☒	
7.32 Have large strings of numbers or letters been broken into chunks?	☐ ☐ ☒	
7.33 Are inactive menu items grayed out or omitted?	☐ ☒ ☐	
7.34 Are there menu selection defaults?	☐ ☐ ☒	
7.35 If the system has many menu levels or complex menu levels, do users have access to an on-line spatial menu map?	☐ ☐ ☒	
7.36 Do GUI menus offer affordance: that is, make obvious where selection is possible?	☒ ☐ ☐	
7.37 Are there salient visual cues to identify the active window?	☒ ☐ ☐	
7.38 Are function keys arranged in logical groups?	☐ ☐ ☒	

7.39 Do data entry screens and dialog boxes indicate when fields are optional?	☐ ☐ ☒	
7.40 On data entry screens and dialog boxes, are dependent fields displayed only when necessary?	☐ ☐ ☒	

8. FLEXIBILITY AND EFFICIENCY OF USE

Accelerators - unseen by the novice user - may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the "average" user (e.g., physical or cognitive ability, culture, language, etc.)

Review Checklist	Yes	No	N/A	Comments
8.1 If the system supports both novice and expert users, are multiple levels of error message detail available?	☐	☐	☒	
8.2 Does the system allow novices to use a keyword grammar and experts to use a positional grammar?	☐	☐	☒	
8.3 Can users define their own synonyms for commands?	☐	☒	☐	
8.4 Does the system allow novice users to enter the simplest, most common form of each command, and allow expert users to add parameters?	☐	☐	☒	
8.5 Do expert users have the option of entering multiple commands in a single string?	☐	☐	☒	
8.6 Does the system provide function keys for high-frequency commands?	☐	☐	☒	
8.7 For data entry screens with many fields or in which source documents may be incomplete, can users save a partially filled screen?	☐	☐	☒	
8.8 Does the system automatically enter leading zeros?	☐	☐	☒	

8.9 If menu lists are short (seven items or fewer), can users select an item by moving the cursor?	☒ ☐ ☐	En toutes les listes existantes dans les interfaces l'utilisateur peut sélectionner un item en déplaçant le curseur.
8.10 If the system uses a type-ahead strategy, do the menu items have mnemonic codes?	☐ ☐ ☒	
8.11 If the system uses a pointing device, do users have the option of either clicking on fields or using a keyboard shortcut?	☐ ☒ ☐	
8.12 Does the system offer "find next" and "find previous" shortcuts for database searches?	☐ ☐ ☒	
8.13 On data entry screens, do users have the option of either clicking directly on a field or using a keyboard shortcut?	☐ ☒ ☐	L'utilisateur a l'option de cliquer directement sur un champ seulement.
8.14 On menus, do users have the option of either clicking directly on a menu item or using a keyboard shortcut?	☐ ☐ ☒	L'utilisateur a l'option de cliquer directement sur un menu seulement.
8.15 In dialog boxes, do users have the option of either clicking directly on a dialog box option or using a keyboard shortcut?	☐ ☐ ☒	L'utilisateur a l'option de cliquer directement sur une fenêtre de dialogue seulement.
8.16 Can expert users bypass nested dialog boxes with either type-ahead, user-defined macros, or keyboard shortcuts?	☐ ☐ ☒	

9. AESTHETIC AND MINIMALIST DESIGN

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Review Checklist	Yes	No	N/A	Comments
9.1 Is only (and all) information essential to decision making displayed on the screen?	☒	☐	☐	Cependant les informations sont présentées d'une façon desorganisée et tortueuse. Le design de l'interface pourrait être retravaillée pour assurer un repérage de l'information plus facile, ce que réduirait sensiblement la surcharge cognitive de l'utilisateur. En fait, juste l'alignement des champs, des textes et des boutons rendrait cette Interface utilisateur graphique (IUG) plus facile de manipuler. "Our brains like order, not chaos" [Hobart95].
9.2 Are all icons in a set visually and conceptually distinct?	☒	☐	☐	Oui mais il faudrait reconcevoir tous les icônes en termes de design car ils ne sont pas représentatifs.
9.3 Have large objects, bold lines, and simple areas been used to distinguish icons?	☐	☒	☐	
9.4 Does each icon stand out from its background?	☒	☐	☐	
9.5 If the system uses a standard GUI interface where menu sequence has already been specified, do menus adhere to the specification whenever possible?	☒	☐	☐	
9.6 Are meaningful groups of items separated by white space?	☐	☒	☐	
9.7 Does each data entry screen have a short, simple, clear, distinctive title?	☐	☒	☐	
9.8 Are field labels brief, familiar, and descriptive?	☒	☐	☐	
9.9 Are prompts expressed in the affirmative, and do they use the active voice?	☐	☐	☒	
9.10 Is each lower-level menu choice associated with only one higher level menu?	☐	☐	☒	
9.11 Are menu titles brief, yet long enough to communicate?	☒	☐	☐	
9.12 Are there pop-up or pull-down menus within data entry fields that have many, but well defined	☒	☐	☐	

entry options?

10. HELP AND DOCUMENTATION

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Review Checklist	Yes	No	N/A	Comments
10.1 If users are working from hard copy, are the parts of the hard copy that go on-line marked?	☐	☐	☒	
10.2 Are on-line instructions visually distinct?	☐	☐	☒	
10.3 Do the instructions follow the sequence of user actions?	☒	☐	☐	
10.4 If menu choices are ambiguous, does the system provide additional explanatory information when an item is selected?	☒	☐	☐	On offre la possibilité à l'apprenant de demander pourquoi on lui pose la question (ce qui l'aide à comprendre ce que le tuteur veut vérifier), comment il peut y répondre (instructions techniques).
10.5 Are data entry screens and dialog boxes supported by navigation and completion instructions?	☒	☐	☐	Il y a des explications au sujet de la navigation et de la façon de compléter les exercices au début mais ces explications seraient utiles en tout temps. Elles pourraient être plus faciles d'accès.
10.6 If menu items are ambiguous, does the system provide additional explanatory information when an item is selected?	☒	☐	☐	Oui, des boîtes contenant des informations supplémentaires apparaissent lorsqu'on passe la souris sur certains items.
10.7 Are there memory aids for commands, either through on-line quick reference or prompting?	☒	☐	☐	
10.8 Is the help function visible; for example, a key labelled HELP or a special menu?	☐	☐	☒	Les fonctions d'aide ne sont pas disponibles dans cette version du prototype.
10.9 Is the help system interface (navigation, presentation, and conversation) consistent with the navigation, presentation, and conversation interfaces of the application it supports?	☐	☐	☒	

10.10 Navigation: Is information easy to find?	☐ ☒ ☐	Les explications théoriques sont faciles à trouver mais il est parfois difficile de se retrouver dans le dialogue avec le tuteur. Il nous est arrivé de constater qu'une partie de la conversation apparaissait <u>sous</u> la zone visible à l'écran alors que nous cherchions les informations dans une fenêtre visible.
10.11 Presentation: Is the visual layout well designed?	☐ ☒ ☐	La mise en page pourrait être retravaillée pour assurer un repérage de l'information plus facile.
10.12 Conversation: Is the information accurate, complete, and understandable?	☒ ☐ ☐	
10.13 Is the information relevant?	☒ ☐ ☐	Oui, l'information présentée est pertinente et aide à poursuivre la démarche mais elle est parfois difficile à trouver.
10.14 Goal-oriented (What can I do with this program?)	☐ ☐ ☒	Les objectifs seront clairement présentés dans ce système et on leur réserve une place importante mais, dans le prototype évalué, on ne peut les consulter.
10.15 Descriptive (What is this thing for?)	☒ ☐ ☐	
10.16 Procedural (How do I do this task?)	☒ ☐ ☐	
10.17 Interpretive (Why did that happen?)	☐ ☒ ☐	On ne comprend pas toujours si on est sur la bonne voie ou non lorsqu'on interagit avec le tuteur.
10.18 Navigational (Where am I?)	☐ ☐ ☒	Nous savons qu'un graphique permettant à l'apprenant de se situer par rapport à l'ensemble du contenu et de voir où il est rendu est prévu mais il n'est pas encore disponible dans cette version du prototype.
10.19 Is there context-sensitive help?	☒ ☐ ☐	L'apprenant peut demander pourquoi on lui pose la question et on le met en contexte.
10.20 Can the user change the level of detail available?	☐ ☐ ☒	
10.21 Can users easily switch between help and their work?	☐ ☐ ☒	
10.22 Is it easy to access and return from the help system?	☐ ☐ ☒	
10.23 Can users resume work where they left off after accessing help?	☒ ☐ ☐	

11. SKILLS

The system should support, extend, supplement, or enhance the user's skills, background knowledge, and expertise - not replace them.

Review Checklist	Yes	No	N/A	Comments
11.1 Can users choose between iconic and text display of information?	☐	☐	☒	
11.2 Are window operations easy to learn and use?	☒	☐	☐	
11.3 If users are experts, usage is frequent, or the system has a slow response time, are there fewer screens (more information per screen)?	☐	☐	☒	
11.4 If users are novices, usage is infrequent, or the system has a fast response time, are there more screens (less information per screen)?	☐	☐	☒	
11.5 Does the system automatically color-code items, with little or no user effort?	☐	☒	☐	Il n'y a pas de gradation dans le niveau de difficulté des objectifs ou des activités et il pourrait être intéressant d'intégrer ces précisions.
11.6 If the system supports both novice and expert users, are multiple levels of detail available.	☐	☐	☒	
11.7 Are users the initiators of actions rather than the responders?	☒	☐	☐	
11.8 Does the system perform data translations for users?	☐	☐	☒	
11.9 Do field values avoid mixing alpha and numeric characters whenever possible?	☐	☐	☒	
11.10 If the system has deep (multilevel) menus, do users have the option of typing ahead?	☐	☐	☒	
11.11 When the user enters a screen or dialog box, is the cursor already positioned in the field users are most likely to need?	☒	☐	☐	
11.12 Can users move forward and backward within a field?	☒	☐	☐	

11.13 Is the method for moving the cursor to the next or previous field both simple and visible?	☒ ☐ ☐	On utilise la souris pour naviguer dans ce système.
11.14 Has auto-tabbing been avoided except when fields have fixed lengths or users are experienced?	☐ ☐ ☒	
11.15 Do the selected input device(s) match user capabilities?	☒ ☐ ☐	
11.16 Are cursor keys arranged in either an inverted T (best for experts) or a cross configuration (best for novices)?	☐ ☐ ☒	
11.17 Are important keys (for example, <u>ENTER</u> , <u>TAB</u>) larger than other keys?	☐ ☐ ☒	
11.18 Are there enough function keys to support functionality, but not so many that scanning and finding are difficult?	☐ ☐ ☒	
11.19 Are function keys reserved for generic, high-frequency, important functions?	☐ ☐ ☒	
11.20 Are function key assignments consistent across screens, subsystems, and related products?	☐ ☐ ☒	
11.21 Does the system correctly anticipate and prompt for the user's probable next activity?	☐ ☐ ☒	Il y a un calcul de probabilité du degré d'atteinte d'un objectif et les questions présentées à l'apprenant vont s'ajuster au modèle de l'apprenant mais pour l'instant, cette individualisation n'est pas programmée.

12. PLEASURABLE AND RESPECTFUL INTERACTION WITH THE USER

The user's interactions with the system should enhance the quality of her or his work-life. The user should be treated with respect. The design should be aesthetically pleasing- with artistic as well as functional value.

Review Checklist	Yes	No	N/A	Comments
12.1 Is each individual icon a harmonious member of a family of icons?	☐	☒	☐	
12.2 Has excessive detail in icon design been avoided?	☒	☐	☐	
12.3 Has color been used with discretion?	☐	☒	☐	Il y a parfois trop de couleurs dans une même page.
12.4 Has the amount of required window housekeeping been kept to a minimum?	☐	☒	☐	Il faut presque toujours réajuster les fenêtres de façon à voir toutes les informations dont on a besoin. Aussi, la fenêtre du menu principal doit constamment être réajustée.
12.5 If users are working from hard copy, does the screen layout match the paper form?	☐	☐	☒	
12.6 Has color been used specifically to draw attention, communicate organization, indicate status changes, and establish relationships?	☒	☐	☐	Oui, mais lorsqu'il y a beaucoup de couleurs pour attirer l'attention sur une même page, c'est moins efficace.
12.7 Can users turn off automatic color coding if necessary?	☐	☒	☐	
12.8 Are typing requirements minimal for question and answer interfaces?	☒	☐	☐	Parfois, on attend une réponse bien précise et ce n'est pas clairement indiqué. La réponse de l'apprenant peut donc être valable sans qu'il le sache.
12.9 Do the selected input device(s) match environmental constraints?	☐	☐	☒	
12.10 If the system uses multiple input devices, has hand and eye movement between input devices been minimized?	☐	☐	☒	
12.11 If the system supports graphical tasks, has an alternative pointing device been provided?	☐	☐	☒	
12.12 Is the numeric keypad located to the right of the alpha key area?	☐	☐	☒	
12.13 Are the most frequently used function keys in the most accessible positions?	☐	☐	☒	
12.14 Does the system complete unambiguous partial input on a data entry field?	☐	☐	☒	

13. PRIVACY

The system should help the user to protect personal or private information- belonging to the user or his/her clients.

Review Checklist	Yes	No	N/A	Comments
13.1 Are protected areas completely inaccessible?	☐	☐	☒	
13.2 Can protected or confidential areas be accessed with certain passwords?	☐	☐	☒	
12.3 Is this feature effective and successful?	☐	☐	☒	