



HAL
open science

Modèles et outils pour favoriser l'articulation entre la généricité d'un assistant intelligent et les spécificités de son usage en EIAH

Le Vinh Thai

► **To cite this version:**

Le Vinh Thai. Modèles et outils pour favoriser l'articulation entre la généricité d'un assistant intelligent et les spécificités de son usage en EIAH. Environnements Informatiques pour l'Apprentissage Humain. Université de Lyon, 2017. Français. NNT : 2017LYSE1096 . tel-01579410

HAL Id: tel-01579410

<https://theses.hal.science/tel-01579410v1>

Submitted on 31 Aug 2017

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

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



N°d'ordre NNT : 2017LYSE1096

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de

l'Université Claude Bernard Lyon 1

École Doctorale InfoMaths N° ED 512

Spécialité de doctorat : Informatique

Soutenue publiquement le 23/06/2017, par :

Le Vinh THAI

Modèles et outils pour favoriser l'articulation entre la généricité d'un assistant intelligent et les spécificités de son usage en EIAH

Devant le jury composé de :

Rapporteurs

Pascal LEROUX
Sébastien IKSAL

Professeur des Universités à l'ENSIM du Mans
Maître de conférences HDR à l'IUT de Laval

Examineurs

Hamid CHAACHOUA
Viviane GUERAUD

Professeur des Universités à l'ESPE Grenoble
Maîtresse de conférences HDR à l'Université Grenoble Alpes

Directrices

Stéphanie JEAN-DAUBIAS
Marie LEFEVRE

Professeure des Universités à l'Université Lyon 1
Maîtresse de conférences à l'Université Lyon 1

UNIVERSITÉ CLAUDE BERNARD - LYON 1

Président de l'Université

Président du Conseil Académique

Vice-président du Conseil d'Administration

Vice-président du Conseil Formation et Vie Universitaire

Vice-président de la Commission Recherche

Directrice Générale des Services

M. le Professeur Frédéric FLEURY

M. le Professeur Hamda BEN HADID

M. le Professeur Didier REVEL

M. le Professeur Philippe CHEVALIER

M. Fabrice VALLÉE

Mme Dominique MARCHAND

COMPOSANTES SANTE

Faculté de Médecine Lyon Est – Claude Bernard

Faculté de Médecine et de Maïeutique Lyon Sud – Charles
Mérieux

Faculté d'Odontologie

Institut des Sciences Pharmaceutiques et Biologiques

Institut des Sciences et Techniques de la Réadaptation

Département de formation et Centre de Recherche en Biologie
Humaine

Directeur : M. le Professeur G. RODE

Directeur : Mme la Professeure C. BURILLON

Directeur : M. le Professeur D. BOURGEOIS

Directeur : Mme la Professeure C. VINCIGUERRA

Directeur : M. X. PERROT

Directeur : Mme la Professeure A-M. SCHOTT

COMPOSANTES ET DÉPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies

Département Biologie

Département Chimie Biochimie

Département GEP

Département Informatique

Département Mathématiques

Département Mécanique

Département Physique

UFR Sciences et Techniques des Activités Physiques et Sportives

Observatoire des Sciences de l'Univers de Lyon

Polytech Lyon

Ecole Supérieure de Chimie Physique Electronique

Institut Universitaire de Technologie de Lyon 1

Ecole Supérieure du Professorat et de l'Education

Institut de Science Financière et d'Assurances

Directeur : M. F. DE MARCHI

Directeur : M. le Professeur F. THEVENARD

Directeur : Mme C. FELIX

Directeur : M. Hassan HAMMOURI

Directeur : M. le Professeur S. AKKOUCHE

Directeur : M. le Professeur G. TOMANOV

Directeur : M. le Professeur H. BEN HADID

Directeur : M. le Professeur J-C PLENET

Directeur : M. Y. VANPOULLE

Directeur : M. B. GUIDERDONI

Directeur : M. le Professeur E. PERRIN

Directeur : M. G. PIGNAULT

Directeur : M. le Professeur C. VITON

Directeur : M. le Professeur A. MOUGNIOTTE

Directeur : M. N. LEBOISNE

Remerciements

Tout d'abord, j'adresse mes remerciements à tous les membres du jury : M. Pascal Leroux, M. Sébastien Ikşal, M. Hamid Chaachoua et Mme. Viviane Guéraud pour vos efforts pour lire mon manuscrit de thèse, assister à ma soutenance en me proposant des questions et des remarques pertinentes et constructives.

Merci à Marie qui m'a encadré jusqu'au bout de ma thèse. Tu es toujours disponible pour m'aider, me donner des conseils très utiles et me transmettre ton expérience. Tu commences et termines les réunions avec le sourire malgré un programme très chargé. Un autre remerciement est adressé à Stéphanie qui m'a donné l'occasion de faire cette thèse. On s'est connus depuis 2012 quand j'ai suivi ton cours pour pouvoir améliorer mon enseignement. J'ai pu ensuite réaliser ma thèse sous ta direction. Tu m'as appris non seulement des connaissances, des compétences, ainsi que la vie française, mais encore la responsabilité. Stéphanie et Marie, vous êtes toujours importantes non seulement pour ma thèse, mais encore pour des recherches futures. Je souhaite donc travailler avec vous à l'avenir. Pour vous remercier, je vous promets d'être un bon enseignant-chercheur sérieux et d'encadrer d'autres étudiants avec les moyens que vous m'avez donnés.

Merci aux membres de LIRIS et de l'Université Lyon 1 avec lesquels j'ai travaillé dans un environnement très confortable, très agréable. En particulier, Blandine une docteure intelligente, sérieuse, tu m'as donné des conseils et de l'aide non seulement pour la thèse, mais encore pour la vie. Je n'oublierai jamais la journée de ski avec toi, que j'aurais pu oublier. Je remercie également Raphaël avec qui j'ai discuté et partagé sur des problèmes techniques. J'espère que tu finiras ta thèse avec grand succès.

Je remercie les responsables de la bourse 911, les enseignants à l'Université des Sciences à Ho Chi Minh ville. Vous m'avez donné les moyens et le support pour ma thèse.

Je remercie mes amis. Où que vous soyez, soit loin au Vietnam, soit près en France, vous êtes toujours à mes côtés. Ma vie est plus belle grâce à vous, mes amis.

Un grand remerciement est réservé pour ma grande famille en France et au Vietnam. Vous êtes toujours dans mon cœur comme je suis toujours dans votre cœur. Mes parents, vous ne pouvez pas comprendre ce texte en français, mais vous comprenez toujours que je vous aime beaucoup. ❤️

Plan du manuscrit

Chapitre 1. Introduction	15
Motivations	16
Le projet AGATE, une réponse aux besoins d'assistance	19
Problématique	20
Scénarios d'usage	21
Résumé des contributions	24
Plan du manuscrit.....	25
Chapitre 2. État de l'art sur les systèmes d'assistance en contexte éducatif	26
2.1. Les types d'assistance	29
2.1.1 Assistance technique.....	29
2.1.2 Assistance pédagogique	30
2.2. Les modes de déroulement d'une assistance	44
2.2.1 Mode indépendant.....	44
2.2.2 Mode successif	45
2.2.3 Mode simultané.....	45
2.2.4 Mode progressif.....	46
2.2.5 Mode interactif.....	46
2.2.6 Classification des modes de déroulement d'une assistance.....	47
2.3. Mise en place d'assistance en contexte éducatif	47
2.3.1 Systèmes conseillers.....	48
2.3.2 Outils auteurs.....	51
2.3.3 Tutorat.....	53
2.3.4 Système de scénarisation pédagogique	56
2.3.5 Hypermédias adaptatifs	61
2.3.6 Grafcet.....	62
2.4. Discussion	64
Chapitre 3. Le projet AGATE, une solution pour ajouter de l'assistance à un logiciel en contexte éducatif.....	67
3.5. Contexte	69
3.5.1 Un processus d'adjonction d'un système d'assistance sur une application existante	70
3.5.2 Le Langage aLDEAS	72

3.5.3	Le système SEPIA	75
3.5.4	Exemple d'un système d'assistance en aLDEAS.....	79
3.6.	Discussion du potentiel de SEPIA en contexte éducatif.....	90
3.6.1	Comment définir le déroulement d'une assistance	92
3.6.2	Comment définir un guidage pédagogique.....	93
3.7.	Approche proposée pour adapter aLDEAS et SEPIA au contexte éducatif ..	95
Chapitre 4.	Proposition d'articulations entre règles aLDEAS.....	97
4.1.	Modèle d'articulation entre règles	99
4.1.1	Mode d'articulation indépendant.....	100
4.1.2	Mode d'articulation successif	102
4.1.3	Mode d'articulation simultané.....	104
4.1.4	Mode d'articulation progressif	106
4.1.5	Mode d'articulation interactif.....	108
4.1.6	Combinaison de modes d'articulation.....	110
4.2.	Processus d'application automatique des contraintes des modes d'articulation aux règles	112
4.2.1	Modèle de contraintes du mode d'articulation sur les règles ou blocs	113
4.2.2	Initialisation des listes.....	115
4.2.3	Identification des contraintes sur les règles	116
4.2.4	Application des contraintes aux règles.....	116
4.2.5	Exemple de mise en œuvre du processus d'application des contraintes.....	117
4.3.	SEPIA-edu : mise en œuvre de l'articulation entre règles	119
4.3.1	Définition de blocs de règles aLDEAS	121
4.3.2	Application des contraintes sur les règles aLDEAS	123
4.4.	Exemple d'un déroulement d'une assistance	125
Chapitre 5.	Proposition de Guidage pédagogique	133
5.1.	Modèle d'activité	134
5.1.1	États de sortie	134
5.1.2	Règles associées à une activité	137
5.2.	Modèle de guidage pédagogique.....	139
5.2.1	Guidage Libre	139
5.2.2	Guidage séquentiel.....	140
5.2.3	Guidage temporel.....	141

5.2.4	Guidage personnalisé	142
5.2.5	Guidage contextuel.....	143
5.3.	Patron de guidage pédagogique	144
5.4.	Processus de transformation d'un guidage pédagogique en règles aLDEAS...	148
5.4.1	Initialisation des listes.....	149
5.4.2	Transformation du guidage pédagogique	149
5.5.	SEPIA-edu : mise en œuvre du guidage pédagogique	152
5.5.1	Identification des activités	153
5.5.2	Définition du guidage pédagogique.....	157
5.5.3	Transformation du guidage pédagogique en règles aLDEAS.....	160
5.5.4	Identification des besoins d'assistance	161
5.5.5	Élaboration de la réponse	161
5.5.6	États de sortie en PMDLé	162
5.5.7	Épi-assistants.....	162
5.6.	Exemple d'un guidage pédagogique défini avec SEPIA-edu	162
Chapitre 6.	Évaluation.....	173
6.1.	Grille d'évaluation	174
6.1.1	Évaluation des modèles du guidage pédagogique et de l'articulation entre règles	174
6.1.2	Évaluation des processus de transformation d'un guidage pédagogique en règles aLDEAS et d'application automatique des contraintes aux règles aLDEAS	176
6.1.3	Évaluation du système SEPIA-edu.....	176
6.2.	Évaluations réalisées	178
6.2.1	Études sur l'assistance.....	178
6.2.2	Analyse des processus de transformation en règles aLDEAS.....	179
6.2.3	Implémentation des modèles et des processus	180
6.2.4	Mise à l'essai de SEPIA-edu en 2015	181
6.2.5	Mise à l'essai de SEPIA-edu en 2017	184
Chapitre 7.	Conclusion et Perspectives	186
	Synthèse	187
	Retours sur les scénarios d'usage.....	191
	Discussion	192
	Perspectives	194

Références	198
Annexes	206
Annexe A Logiciels utilisés et étudiés	207
Annexe B Algorithmes utilisés dans les processus.....	208
B.1 Algorithmes pour le processus d'application automatique des contraintes des modes d'articulation aux règles	208
B.2 Algorithmes pour le processus de transformation du guidage pédagogique en règles aLDEAS.....	211
Annexe C Typologie des éléments aLDEAS.....	212
Annexe D Exemples d'assistance définies et exécutées avec SEPIA-edu.....	215
D.1 Exemples de déroulement d'une assistance	215
D.2 Exemples de guidage pédagogique.....	220
Annexe E Documents pour les mise à l'essai	222
E.1 Mise à l'essai en 2015.....	222
E.2 Mise à l'essai en 2017	235

Table des illustrations

Figure 1 : Vue d'ensemble de notre classification des caractéristiques de l'assistance en contexte éducatif.....	28
Figure 2 : Classification des caractéristiques de l'assistance technique (Ginon, 2014)	29
Figure 3 : Instructions explicitant l'activité dans AMBRE-add (Nogry, Guin and Jean-Daubias, 2008)	31
Figure 4 : Carte d'exercices sur Aplusix	32
Figure 5 : Exemple proposé à l'élève dans AMBRE-add (Nogry, Guin and Jean-Daubias, 2008)	33
Figure 6 : Exemple proposé à l'élève dans (Mathenpoche, 2016).....	33
Figure 7 : Indice proposé par Aplusix.....	34
Figure 8 : Les indices (1,2) et la solution (3) proposée par ActiveMath	35
Figure 9 : Explications proposées par IXL Learning	35
Figure 10 : Explication sur un concept proposée par ActiveMath	36
Figure 11 : Diagnostic proposé par Matou Matheux (Matheux, 2016)	37
Figure 12 : Exemples de bilans proposés par Educ@ffix (Ergun, 2006).....	37
Figure 13 : Scénario et granularité des unités d'apprentissage (Pernin et Lejeune 2004) ...	39
Figure 14 : Exemple d'un guidage libre sur Matou Matheux (Matheux, 2016)	41
Figure 15 : Exemple d'un type de guidage temporel sur SpiralConnect.....	41
Figure 16 : Exemple d'un guidage contextualisé dans Mathenpoche (Mathenpoche, 2016)	42
Figure 17 : La première leçon de la phase d'évaluation de Gymglish (Gymglish, 2016)	43
Figure 18 : Classification des caractéristiques de l'assistance pédagogique	43
Figure 19 : Exemple de l'affichage de la vue d'ensemble d'un cours d'IXL Learning (IXL, 2016)	44
Figure 20 : Exemple d'une étape d'assistance dans le tutoriel intégré à Connectify (Connectify, 2016) et d'une étape de vérification des réponses dans EOLF (EOLF, 2016)	45
Figure 21 : Exemples de vérification de la réponse sur ActiveMath (Melis et al., 2001).....	46
Figure 22 : Classification des modes de déroulement d'une assistance	47
Figure 23 : Édition d'un graphe de tâches dans Epitalk (Paquette, Pachat and Giroux, 1994)	48
Figure 24 : Epitalk confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	49
Figure 25 : Exemple d'un conseil sous forme d'une fenêtre pour EducaSource avec Marco (Richard, 2008)	50
Figure 26 : Marco confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	51
Figure 27 : Exemple de création d'un QCM avec Hot Potatoes (Winke and MacGregor, 2001)	52
Figure 28 : Hot Potatoes confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	52
Figure 29 : Exemple d'un tuteur dans Geometer's sketchpad (Ritter and Koedinger, 1995)	54
Figure 30 : Interaction entre le tuteur et le logiciel (Ritter and Koedinger, 1995).....	54

Figure 31 : Lightweight Tutoring Agents confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif	55
Figure 32 : FORMID confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	56
Figure 33 : OASIF confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	57
Figure 34 : WebSchool confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	58
Figure 35 : Exemple d'un scénario créé par l'éditeur de scénario Telos	59
Figure 36 : Exemple de la création de règles pour une condition dans Telos	59
Figure 37 : Exemple de la création des règles pour un agent conseiller dans Telos.....	60
Figure 38 : Telos confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	60
Figure 39 : Les systèmes de scénarisation pédagogique confrontés à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	61
Figure 40 : L'approche des hypermédias adaptatifs confrontée à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	62
Figure 41 : Éléments principaux dans Grafcet.....	63
Figure 42 : Exemple de la séquence simultanée du Grafcet	63
Figure 43 : Classification complète des caractéristiques d'une assistance en contexte éducatif	65
Figure 44 : Les travaux connexes confrontés à notre approche et aux caractéristiques de l'assistance en contexte éducatif.....	66
Figure 45 : Modèles et outils proposés dans le projet AGATE pour mettre en place des systèmes d'assistance épiphytes	70
Figure 46 : Processus d'adjonction d'un système d'assistance (Ginon, Jean-Daubias and Champin, 2013)	70
Figure 47 : Éléments de base aLDEAS.....	72
Figure 48 : Exemple d'une attente d'un événement composés E_1 « sauvegarder un document Word »	73
Figure 49 : Exemple d'une action composée	74
Figure 50 : Patron de règles aLDEAS.....	74
Figure 51 : Exemple d'une règle sur ActiveMath	75
Figure 52 : Architecture du système SEPIA (Ginon, 2014).....	76
Figure 53 : Écran principal de l'éditeur d'assistance de SEPIA	77
Figure 54 : Ajout d'un bouton et de symboles sur l'interface de EOLF à l'aide de SEPIA	78
Figure 55 : Étapes du lancement d'un système d'assistance selon les deux versions du moteur d'assistance SEPIA	78
Figure 56 : Vue d'ensemble de l'exemple de système d'assistance qui guide l'utilisateur pour la correction de yeux rouges.....	79
Figure 57 : Exécution de l'étape "demande si l'utilisateur souhaite de l'aide" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA	80
Figure 58 : Définition de la règle R_0 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	80
Figure 59 : Exécution de l'étape "demande de clic sur édition" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA.....	81

Figure 60 : Définition de la règle R_1 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	81
Figure 61 : Exécution de l'étape "demande de la sélection d'un dossier" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA	82
Figure 62 : Définition de la règle R_2 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	82
Figure 63 : Définition de la règle R_3 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	83
Figure 64 : Exécution de l'étape "demande plus précise de sélection du dossier" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA	83
Figure 65 : Définition de la règle R_4 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	84
Figure 66 : Exécution de l'étape "demande de sélection d'une photo" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA	85
Figure 67 : Définition de la règle R_5 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	86
Figure 68 : Exécution de l'étape "demande plus précise de sélection d'une photo" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA.....	86
Figure 69 : Définition de la règle R_6 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	87
Figure 70 : Exécution de l'étape "demande de clic sur les outils" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA.....	87
Figure 71 : Exécution de l'étape "demande de clic sur le bouton yeux rouges" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA	88
Figure 72 : Définition de la règle R_7 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	88
Figure 73 : Exécution de l'étape "explication d'utilisation de la fonctionnalité" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA	89
Figure 74 : Définition de la règle R_8 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape	90
Figure 75 : Affiche du message de remerciement à la fin de l'étape "explication d'utilisation de la fonctionnalité" du système d'assistance de PhotoScape avec le moteur SEPIA.....	90
Figure 76 : Exemple de règles aLDEAS permettant une assistance sur ActiveMath selon le mode d'articulation successif	92
Figure 77 : Tableau contenant la liste des règles aLDEAS dans SEPIA	93
Figure 78 : Extrait des règles définissant un guidage pédagogique contextualisé.....	94
Figure 79 : Approche proposée pour adapter aLDEAS et SEPIA au contexte éducatif	95
Figure 80 : Modèle d'articulation entre les règles d'un système d'assistance en aLDEAS .	100
Figure 81 : Exemple d'assistance sur IXL Learning avec trois règles aLDEAS articulées en mode indépendant	101
Figure 82 – Représentation en aLDEAS de n règles en mode d'articulation successif	102
Figure 83 : Exemple du tutoriel de Connectify défini avec trois règles aLDEAS articulées en mode successif.....	104
Figure 84 : Représentation en aLDEAS de n+1 règles en mode d'articulation simultané ..	105
Figure 85 : Exemple du diagnostic des questions sur le site EOLF avec trois règles aLDEAS articulées en mode simultané.....	106

Figure 86 : Représentation en aLDEAS de n règles en mode d'articulation progressif.....	107
Figure 87 : Exemple d'assistance progressive sur un exercice d'ActiveMath avec trois règles articulées en mode progressif	108
Figure 88 : Représentation en aLDEAS de n règles en mode d'articulation interactif	109
Figure 89 : Exemple du diagnostic sur un exercice d'ActiveMath avec trois règles articulées en mode progressif.....	110
Figure 90 : Représentation de la combinaison de trois blocs de règles	111
Figure 91 : Processus d'application des contraintes d'articulation.....	113
Figure 92 : Liste des contraintes liées aux modes d'articulation	115
Figure 93 : Définition d'une assistance avec la version initiale du système SEPIA (Ginon, 2014)	120
Figure 94 : Architecture de l'éditeur de SEPIA-edu adoptant le modèle d'articulation entre règles et le processus d'application automatique des contraintes	121
Figure 95 : Définition d'un bloc de règles articulées en mode successif dans SEPIA-edu ..	122
Figure 96 : Description d'un bloc de règles stockée dans le fichier de description des règles de SEPIA-edu.....	123
Figure 97 : Aperçu des événements générés par l'éditeur d'assistance de SEPIA-edu pour mettre en œuvre l'articulation entre les blocs d'un système d'assistance.....	124
Figure 98 : Message de notification de contrainte non-respectée dans SEPIA-edu	125
Figure 99 : Vue d'ensemble du système d'assistance qui ajoute un bouton d'aide et propose progressivement une explication et un exemple lors des clics sur ce bouton d'aide	125
Figure 100 : Exécution de l'étape "ajout d'un bouton d'aide" du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu.....	126
Figure 101 : Définition de la règle R_9 avec l'éditeur SEPIA-edu pour le système d'assistance de MatouMatheux.....	126
Figure 102 : Définition du bloc de règles B_1 pour MatouMatheux avec l'éditeur SEPIA-edu	127
Figure 103 : Génération de la règle R_{G_0} (invisible pour les concepteurs d'assistance) avec l'éditeur SEPIA-edu pour le système d'assistance de MatouMatheux.....	127
Figure 104 : Exécution de l'étape "affichage d'une explication" du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu.....	128
Figure 105 : Définition de la règle R_9 avec l'éditeur SEPIA-edu pour le système d'assistance MatouMatheux.....	128
Figure 106 : Définition du bloc de règles B_2 avec l'éditeur SEPIA-edu pour le système d'assistance MatouMatheux.....	129
Figure 107 : Exécution de l'étape "affichage d'un exemple" du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu.....	129
Figure 108 : Définition de la règle R_{10} avec l'éditeur SEPIA-edu pour le système d'assistance MatouMatheux.....	130
Figure 109 : Exécution de l'étape "remplissage automatique de l'opérande" du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu	130
Figure 110 : Définition de la règle R_{11} avec l'éditeur SEPIA-edu pour le système d'assistance MatouMatheux.....	131
Figure 111 : Exécution de l'étape "remplissage automatique de la puissance" du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu	131

Figure 112 : Définition de la règle R ₁₂ avec l'éditeur SEPIA-edu pour le système d'assistance de MatouMatheux.....	132
Figure 113 : Définition des états de sortie pour SEPIA-edu en PMDLe	136
Figure 114 : Exemple d'une règle aLDEAS permettant de mettre fin à une activité en spécifiant ses états de sortie.....	138
Figure 115 : Modèle de guidage libre	140
Figure 116 : Modèle de guidage séquentiel	141
Figure 117 : Modèle de guidage temporel	142
Figure 118 : Modèle de guidage personnalisé	143
Figure 119 : Guidage contextuel	144
Figure 120 : Patron de guidage pédagogique	145
Figure 121 : Exemple de guidage pédagogique défini avec le patron de guidage pédagogique	146
Figure 122 : Traduction du patron de guidage pédagogique en règles aLDEAS	148
Figure 123 : Processus de transformation d'un guidage pédagogique en règles aLDEAS..	149
Figure 124 : Règle R ₀ générée dans l'étape de « création de la règle de proposition des activités »	150
Figure 125 : Règle R ₀ mise à jour et règles R ₁₁ et R ₁₂ générées	151
Figure 126 : Règle R ₀ mise à jour et règles R ₂₁ , R ₂₂ générées.....	151
Figure 127 : Architecture de l'éditeur de SEPIA-edu adoptant le modèle d'activité, le patron de guidage pédagogique et le processus de transformation du guidage pédagogique en règles aLDEAS.	153
Figure 128 : Identification d'une activité dans SEPIA-edu	154
Figure 129 : Définition d'un état de sortie d'une activité dans SEPIA-edu.....	155
Figure 130 : Définition d'une échelle dans SEPIA-edu respectant le langage PMDLe	155
Figure 131 : Définition d'une règle et association de cette règle à l'activité dans SEPIA-edu	156
Figure 132 : Description des activités stockée dans le fichier de description des règles aLDEAS de SEPIA-edu.....	156
Figure 133 : Description de l'association des règles aux activités dans le fichier de description des règles aLDEAS de SEPIA-edu.....	157
Figure 134 : Écran de définition d'un guidage pédagogique.....	158
Figure 135 : Illustration de la mise en œuvre de la condition « et/ou » du patron de guidage dans l'éditeur de SEPIA-edu	159
Figure 136 : Description d'un guidage pédagogique stockée dans le fichier de description des règles aLDEAS de SEPIA.	160
Figure 137 : Extrait des règles générées par l'éditeur d'assistance de SEPIA-edu.....	161
Figure 138 : Extrait des conditions générées dans l'éditeur d'assistance de SEPIA-edu	161
Figure 139 : Exemple du message dynamique qui affiche une liste d'activités dans SEPIA-edu	162
Figure 140 : Vue d'ensemble du système d'assistance qui propose à l'apprenant d'une part une liste d'activités organisées dans un guidage pédagogique et d'autre part de l'assistance à la réalisation d'une activité sélectionnée.....	163
Figure 141 : Exécution de l'étape « guider l'apprenant vers l'activité » de l'activité A ₁ du système d'assistance du MOOC « Racines » avec la version Web du moteur SEPIA-edu.....	163

Figure 142 : Identification de l'activité A ₁ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines ».....	164
Figure 143 : Définition de R ₇ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »	165
Figure 144 : Définition de R ₈ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »	165
Figure 145 : Exécution de l'étape « guider l'apprenant à l'activité » de l'activité A ₂ du système d'assistance du MOOC « Racines » avec la version Web du moteur SEPIA-edu.....	165
Figure 146 : Identification de l'activité A ₂ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines ».....	167
Figure 147 : Définition de la règle R ₁₁ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines ».....	167
Figure 148 : Définition de la règle R ₁₂ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines ».....	167
Figure 149 : Exécution de l'étape « guider l'apprenant à l'activité » de l'activité A ₂ du système d'assistance du MOOC « Racines » avec la version Web du moteur SEPIA-edu.....	168
Figure 150 : Identification de l'activité A ₃ dans l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines ».....	169
Figure 151: Définition de la règle R ₁₃ dans l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines ».....	169
Figure 152 : Définition de la règle R ₁₄ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines ».....	169
Figure 153 : Exécution du guidage pédagogique « Affichage une liste d'activités » du système d'assistance du MOOC « Racines » avec la version Web du moteur SEPIA-edu.....	170
Figure 154 : Définition du guidage pédagogique G ₁ pour le système d'assistance du MOOC « Racines » avec l'éditeur SEPIA-edu.....	171
Figure 155 : Génération de règles avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »	171
Figure 156 : Résultats relatifs à la réussite pour la compréhension et la modification du système d'assistance selon les trois méthodes de définition	182
Figure 157 : Résultats relatifs à la préférence les concepteurs pour créer un nouveau système d'assistance	182
Figure 158 : Résultats relatifs à la préférence d'une méthode par les concepteurs pour comprendre (à gauche) et définir (à droite) un système d'assistance	183
Figure 159 : Résultats relatifs à la difficulté de compréhension d'un système d'assistance	183
Figure 160 : Résultats relatifs à la difficulté de définition d'un système d'assistance	183
Figure 161 : Synthèse de nos contributions pour adapter SEPIA au contexte éducatif	188
Figure 162 : Système SEPIA-edu constitué par le système SEPIA et la surcouche métier ..	193
Figure 163 : Exemple de règle permettant un diagnostic dans SEPIA	197
Figure 164 : Logiciels étudiés pour classifier les caractéristiques de l'assistance en contexte éducatif	207
Figure 165 : Typologie des événements élémentaires aLDEAS.....	213
Figure 166 : Typologie des actions d'assistance élémentaires aLDEAS	214
Figure 167 : Écran SEPIA-edu de définition du tutoriel sur NetBeans	215

Figure 168 : Écran SEPIA-edu de définition des instructions explicitant l'activité sur GeoGebra avec SEPIA-edu	216
Figure 169 : Écran SEPIA-edu de définition de l'assistance sur IXL Learning.....	217
Figure 170 : Écran SEPIA-edu de définition de l'assistance sur Matou Matheux.....	217
Figure 171 : Écran SEPIA-edu de définition du tutoriel sur Connectify	218
Figure 172 : Écran SEPIA-edu de définition du diagnostic sur EOLF	219
Figure 173 : Écran SEPIA-edu de définition de l'assistance progressive sur ActiveMath ...	219
Figure 174 : Écran SEPIA-edu de définition d'un guidage pédagogique sur le MOOC Fun « Racines »	220
Figure 175 : Écran SEPIA-edu de définition d'un guidage pédagogique sur ASKER	221
Figure 176 : création d'un guidage pédagogique.....	237
Figure 177 : Création d'une activité.....	238

Chapitre 1. Introduction

Cette thèse s'inscrit dans le cadre des recherches en EIAH (Environnements Informatiques pour l'Apprentissage Humain) dont l'objectif est de susciter, accompagner et personnaliser l'apprentissage. Le champ de recherche des EIAH est un domaine pluridisciplinaire à la croisée de l'informatique, de la psychologie cognitive, de la didactique et des sciences de l'éducation. Dans cette thèse en informatique, nous étudions plus particulièrement l'assistance à l'utilisateur d'applications informatiques en contexte éducatif. Elle est réalisée au sein du projet AGATE (AGATE, 2016).

Motivations

Dans le contexte éducatif, de plus en plus d'applications informatiques sont utilisées. L'apprenant est, avec l'enseignant, l'un des utilisateurs principaux de ces applications informatiques. Les situations d'apprentissage avec des applications informatiques sont variées. Dans le cadre de cette thèse, nous nous intéressons à l'utilisation individuelle des applications par un apprenant.

Les difficultés de prise en main et d'utilisation, ainsi que le manque de feedback pédagogique et de guidage pédagogique dans certaines applications informatiques utilisées par les apprenants en contexte éducatif montrent un besoin d'assistance. Pour répondre à ce besoin, la mise en place d'une assistance est des solutions existantes.

Besoins d'assistance

Les applications utilisées en contexte éducatif sont de deux types.

Ces applications peuvent être des logiciels **pédagogiques**. Il peut s'agir de logiciels conçus dans le cadre de recherches en EIAH, de logiciels conçus par des entreprises privées ou conçus directement par des enseignants pour répondre à leurs besoins. Par exemple, Cabri-Géomètre (Laborde and Capponi, 1994) est un logiciel issu de recherches en EIAH qui permet aux apprenants de construire des figures géométriques ; le site Maxicours (Maxicours, 2016) développé par la société EDUCLEVER (EDUCLEVER, 2016) fournit des cours et des exercices de soutien scolaire aux apprenants ; le site English OnLine France (EOLF, 2016) fournit des exercices sous forme de QCM créés par des enseignants.

En parallèle de ces logiciels pédagogiques, certaines applications « **non-pédagogiques** », c'est-à-dire non spécifiquement conçues pour l'apprentissage, sont également utilisées en contexte éducatif. Dans ce cas, une activité d'apprentissage peut avoir pour objectif la découverte ou la maîtrise d'un outil informatique. Par exemple, des étudiants en informatique qui suivent un cours de programmation en C++ apprennent à utiliser l'environnement de développement Visual Studio en TP. Dans cet exemple, la maîtrise de ce logiciel est une compétence à acquérir par les étudiants. Par ailleurs, certaines activités d'apprentissage dont l'objectif n'est pas la maîtrise d'un outil informatique nécessitent également l'utilisation d'applications non-pédagogiques. Ainsi, dans le cadre d'un cours de bureautique, un enseignant peut faire utiliser Word (Microsoft, 2016) à ses élèves pour apprendre à présenter des lettres officielles ; dans le cadre d'un cours de mathématiques, un enseignant peut faire utiliser Excel (Microsoft, 2016) à ses élèves comme outil d'aide à la résolution de problèmes d'algèbre (Ritter and Koedinger, 1995) ; un formateur en langues étrangères peut également faire utiliser à ses stagiaires un logiciel d'enregistrement de son pour leur faire travailler la prononciation. Dans ces différents contextes, la non-maîtrise de l'application « non-pédagogique » utilisée comme support de l'activité peut freiner l'acquisition des connaissances pédagogiques en jeu. Enfin, dans certains cas, les enjeux d'une activité pédagogique peuvent porter à la fois sur des connaissances relatives à une application non-pédagogique et sur d'autres connaissances relatives au domaine étudié. Par exemple, dans le cadre d'un cours d'informatique, un enseignant peut imposer à ses étudiants d'utiliser un environnement de développement particulier, par exemple NetBeans (Netbeans, 2014), pour réaliser des travaux

pratiques dans le but d'apprendre à utiliser cet environnement tout en acquérant des connaissances en programmation.

Quelles que soient les applications informatiques prescrites, pédagogiques ou non-pédagogiques, les apprenants peuvent renoncer à leur utilisation en raison de difficultés de prise en main et d'utilisation (Cordier *et al.*, 2010). Dans le contexte éducatif, l'obligation d'utiliser ces logiciels pour l'apprentissage et leur difficulté de prise en main peuvent aller jusqu'à compromettre l'acquisition des connaissances. Pour éviter cet écueil, une solution consiste à fournir une **assistance technique** aux apprenants.

De plus, concernant l'activité d'apprentissage en elle-même, certaines applications ne proposent pas de feedbacks pédagogiques ou d'instructions explicitant l'activité d'apprentissage à réaliser. Concernant le choix d'une activité pertinente pour un apprenant donné parmi un ensemble d'activités, celui-ci s'avère souvent difficile, que ce soit pour l'apprenant ou pour son enseignant. Pour faciliter ce choix, une solution consiste à proposer un guidage pédagogique, mais toutes les applications ne le proposent pas. Ainsi, le site de Matou Matheux (Matou Matheux, 2016) met à disposition des enseignants et de leurs élèves un ensemble d'exercices. Cependant, le site ne propose pas de guidage pédagogique qui orienterait l'apprenant vers des activités adaptées à ses besoins. De plus, dans un exercice, Matou Matheux fournit comme seul retour pédagogique un diagnostic de la réponse sous forme d'une indication de réussite ou d'échec. On pourrait imaginer qu'un enseignant faisant travailler ses élèves sur ce site souhaite un retour plus riche, comme l'affichage d'un indice pour débloquent un élève en difficulté ou la proposition d'un exemple similaire à l'exercice en cours avant de donner la correction de celui-ci.

Ces manques existent bien sûr également dans de nombreuses applications non-pédagogiques, dans une plus grande mesure qui plus est. Ces applications ne proposent en effet quasiment pas d'activités d'apprentissage, puisque par définition elles ne sont pas conçues pour l'apprentissage. Le guidage pédagogique est donc inexistant. Si l'on reprend nos exemples précédents, Excel ne propose aucune activité d'apprentissage en mathématiques, et Visual Studio ne propose pas d'activité d'apprentissage pour acquérir des connaissances en programmation. De plus, les logiciels non-pédagogiques sont conçus pour permettre la réalisation d'une tâche, souvent le plus efficacement possible. Ils cherchent donc à faciliter l'utilisation, par exemple en fournissant les solutions aux problèmes rencontrés ou en effectuant les actions à la place de l'utilisateur. Ainsi, ces situations ne fournissent pas de feedback pédagogique approprié.

Ces différentes situations où l'on souhaite enrichir des applications informatiques existantes avec des compléments pédagogiques nécessitent une **assistance pédagogique**.

En combinant **assistance technique** et **assistance pédagogique** pour compléter les logiciels pédagogiques ou non-pédagogiques utilisés par les apprenants, il est possible d'apporter une solution complète pour dépasser les limites existantes des applications utilisées en contexte éducatif.

Mise en place de l'assistance

L'enseignant est souvent utilisateur secondaire des applications sur lesquelles travaillent ses apprenants. Dans ce rôle, il souhaite généralement fournir aux apprenants de l'assistance technique et pédagogique. De nombreux exemples montrent que les enseignants savent mettre en place de telles assistances, que ce soit ou non avec l'aide d'applications informatiques.

Ainsi, lorsque des enseignants demandent à leurs apprenants de travailler sur des applications informatiques, ils les guident lors de l'utilisation de ces applications. Cela peut passer par l'élaboration de vidéos ou de documents d'explication ou par la configuration du système avant son utilisation par les apprenants. Par exemple, dans un cours d'IHM (Interactions Homme-Machine) de seconde année de Licence Informatique à l'Université Lyon 1, des professeurs utilisent le logiciel de capture vidéo Camtasia Studio (Camtasia, 2016), une application non-pédagogique, pour créer 5 tutoriels sous forme d'une vidéo de démonstration sur NetBeans. Ces tutoriels sont utilisés lors du premier TP de l'UE d'IHM consacré à la programmation avec NetBeans. La création d'un tutoriel est un moyen pour ces enseignants, d'une part d'aider leurs élèves à prendre en main et à utiliser l'application non-pédagogique NetBeans, et d'autre part d'explicitier l'activité d'apprentissage, non prévue par NetBeans, qu'ils proposent aux étudiants.

Par ailleurs, les enseignants donnent très souvent des feedbacks pédagogiques aux apprenants en difficulté pour les débloquer. Par exemple, l'application Hot Potatoes (Winke and MacGregor, 2001) permet à l'enseignant de créer des QCM (questionnaires à choix multiples) sous forme d'une page Web. Pour proposer un feedback pédagogique, les enseignants configurent l'outil en précisant les bonnes réponses et en donnant des explications sur les mauvaises réponses. Ce feedback est ensuite fourni aux apprenants lors de demandes de vérification des réponses.

Concernant le guidage pédagogique, les enseignants proposent eux-mêmes des activités d'apprentissage adaptées à la situation d'apprentissage de l'apprenant. Certaines applications aident l'enseignant dans l'exploitation pédagogique d'outils non-pédagogiques, par exemple par des outils de création d'un guidage pédagogique qui propose une séquence d'activités à suivre par l'apprenant. Par exemple, l'EIAH OASIF (Galison and Nouveau, 2003) aide l'enseignant à planifier les activités d'apprentissage. De même, d'autres applications non-pédagogiques telles que Word peuvent être utilisées pour élaborer un plan d'activités d'apprentissage.

En dehors de la mise en place d'une assistance technique et pédagogique sous forme d'applications, le dispositif le plus courant dans l'éducation est l'aide humaine apportée directement par les enseignants en passant derrière chaque apprenant. Ils peuvent aider directement à utiliser les applications, fournir des feedbacks pédagogiques, dans certains cas, ils peuvent diriger l'apprenant vers d'autres activités plus pertinentes. Cependant, cette manière n'est pas optimale car elle exige la présence permanente de l'enseignant au côté des apprenants, ce qui pose fréquemment des problèmes de sur-sollicitation de l'enseignant (Leroux, 1993; Paquette, 1999; Leroux and Vivet, 2000). Pour aider les enseignants dans leur tâche de suivi des apprenants et d'aide à ces apprenants, de nombreux travaux sur le tutorat fournissent des outils support à l'enseignant (Gounon, Leroux and Dubourg, 2004; De Lièvre, Depover and Dillenbourg, 2006; Laperrousz, Leroux and Teutsch, 2007; Quintin, 2007). Ces

outils peuvent aller jusqu'à proposer une assistance informatique automatique apportée aux apprenants et permettre donc de décharger l'enseignant de ses interventions (Labat, Pernin and Guéraud, 2006).

Pour répondre à ce besoin d'assistance des apprenants auxquels doivent faire face les enseignants, une solution précoce dans le cycle de vie d'une application est de modifier le code source de celle-ci. Mais ce code source n'est pas toujours disponible et quand il l'est, il n'est pas facile pour des enseignants, qui ne sont pas informaticiens, de le modifier, car cela nécessiterait des compétences en programmation qu'ils n'ont pas forcément. Dans un tel contexte éducatif, il est donc pertinent de mettre en place de l'assistance qui n'exige pas de modifier le code source de l'application à assister ni de compétences en programmation. Pour cela, la mise en place de **systèmes d'assistance épiphytes** est une solution appropriée. Un système d'assistance épiphyte est un système d'assistance destiné aux utilisateurs d'une application-cible donnée qui peut être greffé sur cette application sans perturber son fonctionnement (Paquette *et al.*, 1996).

Pour que les enseignants puissent mettre en place un tel système d'assistance sans programmation, il est nécessaire de trouver un **moyen** facilitant la création et l'exécution de cette assistance de façon épiphyte sur les applications-cibles. De plus, les applications informatiques sont techniquement de types variés : Windows, Java, Web, etc. L'impossibilité d'accéder au code source et la multiplicité des techniques rendent souhaitable la mise en place d'un **moyen unifié**, prenant en charge la diversité des types d'applications.

En plus de ces difficultés techniques, chaque application utilisée en contexte éducatif peut couvrir différents domaines de connaissances : mathématiques (Matou Matheux, ActiveMath), informatique (tortue logo, NetBeans), langue (EOLF), etc. Ces domaines peuvent être très larges. Aussi, ils sont structurés en sous-domaines (algèbre linéaire, géométrie, etc. par exemple pour les mathématiques) et s'adressent à différents niveaux d'apprentissage (primaire, secondaire, etc.). Il faut donc un moyen unifié permettant de mettre en place l'assistance en tenant compte de ces diversités. Ce moyen ne doit donc pas être spécifique à un domaine donné. Cela rend souhaitable l'adoption d'une **approche générique**.

Pour toutes ces raisons, nous défendons l'idée qu'une solution pour assister les apprenants lors de l'utilisation d'application informatique est d'offrir un **moyen unifié** permettant à l'enseignant de mettre en place une **assistance technique** et **pédagogique** sur les applications existantes en adoptant une **approche épiphyte et générique**. Ce moyen ne doit pas exiger de **compétences en programmation**, mais seulement des connaissances en pédagogie, pour que les enseignants concentrent leurs interventions sur le contenu pédagogique.

Le projet AGATE, une réponse aux besoins d'assistance

Le projet AGATE propose des modèles et des outils qui peuvent répondre aux exigences que nous avons identifiées ci-dessus. Ainsi, le **système SEPIA** (Ginon, Jean-Daubias, *et al.*, 2014b), développé dans le cadre de ce projet, permet la mise en place des systèmes d'assistance de **façon épiphyte** sur les applications-cibles. SEPIA n'est spécifique ni à un **domaine d'application** ni à un **type d'applications** (Windows, Java, Web, etc.). Au sein de ce système, la définition d'un

système d'assistance se fait grâce à un langage graphique, le langage aLDEAS. Cet aspect graphique permet au concepteur de l'assistance de concevoir des systèmes d'assistance complets sans **exiger de compétences en programmation**. Le système SEPIA et le langage aLDEAS ont été proposés pour permettre la définition d'assistances techniques afin de pallier les difficultés de prise en main et d'utilisation des applications-cibles.

Dans le cadre de cette thèse, nous souhaitons donc repartir de ce travail en raison de ses avantages adaptés aux besoins que nous avons identifiés : approche épiphyte, unifiée, générique, aucune compétence requise en programmation. Néanmoins, nous devons étudier comment exploiter et compléter ce travail pour permettre de proposer de l'assistance pédagogique et non seulement technique. Cela conduit aux questions de recherche que nous présentons de la section suivante.

Problématique

La problématique générale de cette thèse peut s'exprimer de la manière suivante :

**Comment mettre en place un système d'assistance épiphyte
en contexte éducatif en adoptant une approche générique ?**

Pour répondre à cette question, nous avons travaillé en deux temps.

Étude d'assistances existantes au sein d'applications utilisées en contexte éducatif.

L'objectif de cette étape était d'identifier les types d'assistances actuellement proposées dans les applications informatiques et la façon dont ces assistances sont proposées. Nous avons donc étudié un grand nombre de logiciels, ainsi qu'effectué un état de l'art afin de proposer, une classification des caractéristiques de l'assistance pouvant être proposée en contexte éducatif. Cette classification détaille les types d'assistance et les différents modes de déroulement d'une assistance en contexte éducatif. Elle est présentée en synthèse de l'état de l'art, dans le Chapitre 2 du manuscrit.

Exploitation et enrichissement des modèles et outils du projet AGATE pour les adapter au contexte éducatif.

L'objectif de cette étape était d'exploiter les propositions faites précédemment au sein du projet AGATE pour proposer une solution à notre problématique et donc obtenir un moyen permettant aux enseignants de mettre en place des systèmes d'assistance épiphyte en contexte éducatif en adoptant une approche générique.

Les modèles proposés au sein du projet AGATE et leur mise en œuvre dans le système SEPIA permettent la mise en place de systèmes d'assistance épiphytes, mais ils avaient pour vocation de permettre la mise en place d'assistance pour des problèmes techniques (c'est-à-dire utilisation et prise en main d'une application).

Nous avons donc tout d'abord vérifié que SEPIA permettait de définir effectivement un système d'assistance en contexte éducatif, c'est-à-dire un système d'assistance prenant en compte les difficultés techniques et pédagogiques des apprenants. Cette étude nous a permis d'identifier une première limite à l'utilisation de SEPIA en contexte éducatif : il n'était pas possible de mettre en place avec SEPIA tous les types d'assistance pédagogique identifiés.

Ensuite, pour les assistances pédagogiques qu'il était possible de mettre en place, nous avons étudié la complexité de définition de tels systèmes d'assistance avec SEPIA. Cette étude nous a permis de faire des propositions pour faire évoluer le système SEPIA et le langage aLDEAS dans le but de faciliter la définition d'un système d'assistance par des concepteurs pédagogiques. Ces concepteurs peuvent être des enseignants, mais plus vraisemblablement des ingénieurs pédagogiques qui mettront en place les assistances souhaitées par les enseignants.

La confrontation de SEPIA aux systèmes d'assistance que nous voulions permettre de proposer en contexte éducatif est présentée dans le Chapitre 3 et les solutions proposées pour, d'une part faciliter la mise en place de systèmes d'assistance, et d'autre part compléter SEPIA, sont présentées dans le Chapitre 4 et le Chapitre 5 du manuscrit.

Scénarios d'usage

Dans cette section, nous présentons quatre scénarios d'usage dans lesquels l'enseignant veut combler les lacunes d'une application en y ajoutant une assistance pédagogique.

Ces scénarios mettent en jeu des **situations variées** dans lesquelles les applications informatiques impliquées ne permettent pas de répondre aux besoins particuliers des acteurs du scénario. Ainsi, ils concernent respectivement le manque de retours pédagogiques, le manque d'instructions explicitant l'activité d'apprentissage, l'absence d'activités d'apprentissage dans l'application, et enfin l'insuffisance de guidage pédagogique au sein d'un ensemble d'activités d'apprentissage.

Ces scénarios s'appuient sur des **logiciels de types variés** puisque (1) certains concernent l'utilisation de logiciels pédagogiques et d'autres l'utilisation de logiciels non-pédagogiques et que (2) certains concernent des applications Web et d'autres des applications dites « bureau » au sens où elle doit être installées en local sur un ordinateur. Ainsi, Matou Matheux (Matou Matheux, 2016) est un site Web qui propose un ensemble d'exercices de mathématiques organisés selon plusieurs niveaux (du primaire au secondaire). GeoGebra (GeoGebra, 2016) est quant à lui un logiciel de mathématiques qui associe géométrie, algèbre, tableur, statistiques et calcul infinitésimal et qui permet aux enseignants de créer des ressources sur lesquels leurs élèves peuvent travailler. Actuellement, il y a plus 800.000 ressources créées sur le site de GeoGebra. NetBeans est un environnement de développement d'application, c'est-à-dire un logiciel non-pédagogique, mais il peut être utilisé en classe pour l'apprentissage de la programmation. Enfin, ASKER (ASKER 2016) est un site Web proposant des exercices d'auto-évaluation utilisés en autonomie par les étudiants.

Parmi les quatre scénarios proposés, deux scénarios sont inspirés de la pratique des enseignants de l'Université Lyon 1. Ainsi, le scénario basé sur NetBeans est inspiré d'une expérimentation menée dans le cadre du projet AGATE dans un cours d'IHM en Licence Informatique, résumée dans une vidéo de démonstration disponible sur le site Web du projet AGATE¹. Le scénario basé sur ASKER est inspiré du besoin de fournir un guidage pédagogique pour le cours « Programmation fonctionnelle et récursive » en Licence Mathématiques et

¹ Projet AGATE : <http://liris.cnrs.fr/agate/>

Informatique. Les deux autres scénarios ont été imaginés pour montrer la variété des possibilités d'exploitation de SEPIA en contexte éducatif.

Scénario 1 : proposer un retour pédagogique pour un exercice de Matou Matheux

Stéphane est enseignant de collège en mathématiques. Il propose à ses élèves de troisième une activité sur la proportionnalité avec Matou Matheux. Ce site Web propose des exercices de mathématiques du niveau primaire au niveau secondaire. Stéphane veut ajouter à cette activité en ligne des feedbacks pédagogiques quand ses élèves rencontrent des difficultés. Il souhaite que ces feedbacks fournissent une explication et un exemple sur l'exercice, fournis progressivement sous forme de messages et de flèches ajoutés à la page Web pour mettre en valeur les composants concernés.

Cependant, Matou Matheux ne propose pas de tels feedbacks pédagogiques. Stéphane doit donc passer voir les élèves chaque fois qu'ils rencontrent une difficulté. Cela demande beaucoup d'énergie à Stéphane. De plus, certains élèves, pourtant en difficulté, n'appellent pas leur enseignant et restent parfois bloqués trop longtemps dans cette activité.

Stéphane souhaite donc enrichir le site Web d'un système d'assistance qui fournirait ces feedbacks de façon progressive directement à l'interface de Matou Matheux. Cela pourrait prendre la forme d'un bouton d'aide sur lequel l'apprenant cliquerait pour demander de l'assistance lorsqu'il rencontre une difficulté. Lors de la première demande d'aide d'un élève, une explication serait proposée sous forme d'un message. Lors de la deuxième demande d'aide sur le même élément, un exemple serait proposé sous forme de messages et de flèches pointant sur les composants concernés de l'interface. Ces messages et flèches seraient affichées les uns après les autres.

Scénario 2 : compléter une activité d'apprentissage de GeoGebra

Dans sa classe de sixième, Stéphane demande aux élèves de réaliser une activité sur la version Web de GeoGebra. GeoGebra est un logiciel de mathématiques, adapté à tous les niveaux scolaires de l'école élémentaire à l'université, qui associe géométrie, algèbre, tableur, statistiques et calcul infinitésimal en un unique logiciel simple à utiliser. L'activité proposée par Stéphane à ses sixièmes est consacrée à la manipulation d'objets géométriques dans le but de maîtriser la propriété : « Deux droites perpendiculaires à une même droite sont parallèles ». Stéphane voudrait expliciter cette activité au sein de GeoGebra pour que ses élèves puissent mieux comprendre comment procéder pour réaliser cette activité.

Cependant, la page Web dans laquelle les élèves travaillent ne comporte pas d'instructions explicitant l'activité à réaliser. Stéphane doit donc fournir aux élèves un document extérieur à GeoGebra explicitant cette activité. Les élèves peuvent alors rencontrer des difficultés à basculer entre GeoGebra et le document explicatif, qu'il soit sous forme électronique ou sous forme papier. De plus, s'ils ne suivent pas strictement les instructions du document, cela peut donner lieu à des incohérences entre l'état de GeoGebra et les consignes données. Une autre solution consiste pour Stéphane à expliciter cette activité oralement. Cependant, certains élèves peuvent ne pas suivre toutes ses instructions, qui n'arrivent pas forcément au moment adapté pour tous.

Stéphane souhaite donc créer dans le site Web de GeoGebra un système d'assistance comportant des instructions sous forme de messages successifs. Ces instructions expliciteraient chaque étape de construction des objets géométriques : construction d'une première droite

sans contrainte, puis construction de deux autres droites perpendiculaires à la première, pour ensuite pouvoir démontrer que les deux dernières droites sont parallèles.

Scénario 3 : proposer une activité pédagogique sur NetBeans

NetBeans est un environnement de développement intégré permettant la programmation en différents langages. Hélène, enseignante en informatique, utilise cet environnement dans son cours d'IHM destiné à des étudiants de 2^{ème} et 3^{ème} de licence d'informatique. Dans leur premier TP de programmation événementielle, les étudiants doivent développer en Java dans NetBeans une application simple. Hélène voudrait faciliter cette première séance pour les étudiants ne maîtrisant pas préalablement NetBeans en proposant une activité *via* un tutoriel sur la programmation événementielle dans NetBeans.

Cependant, NetBeans étant une application non-pédagogique, il ne dispose pas d'une telle activité d'apprentissage pour faire travailler les étudiants. Hélène crée donc son tutoriel sous forme d'une vidéo guidant les étudiants. Il s'avère alors souvent difficile pour les étudiants de basculer entre NetBeans et la vidéo qui diffuse le tutoriel sans interruption. Ainsi, les étudiants doivent suspendre l'utilisation de NetBeans pour regarder la vidéo. À l'inverse, ils doivent également mettre en pause la vidéo quand ils utilisent NetBeans pour ne pas manquer les instructions suivantes.

Hélène souhaite donc créer un système d'assistance dans lequel un tutoriel est affiché directement sur l'interface de NetBeans. Ce tutoriel se composerait de messages et de flèches pour guider les étudiants dans la création d'un projet NetBeans, la manipulation de ses propriétés, la création d'une fenêtre simple et la création des fonctions traitant les événements sur l'interface. Ce tutoriel se déclencherait en fonction des actions des apprenants à l'interface de NetBeans.

Scénario 4 : proposer un guidage pédagogique au sein d'une plateforme d'exercices d'auto-évaluation

ASKER est une plateforme proposant des exercices d'auto-évaluation. Marie, enseignante en informatique à l'université, enseigne la programmation fonctionnelle et récursive en licence d'informatique. Dans le cadre de son cours, les étudiants de 1^{ère} année de Licence utilisent ASKER pour s'entraîner à la maison et identifier les notions non comprises en cours avant de venir en travaux dirigés. Ils réalisent tout ou partie des exercices proposés sur ASKER. Marie voudrait ajouter un guidage pédagogique qui organise les activités pour les donner dans un certain ordre aux étudiants au lieu de les laisser gérer l'ordre dans lequel ils font ces activités. Cet ordre doit être adapté à l'avancement du cours et aux lacunes de chaque étudiant.

Cependant, ASKER ne lui permet pas de créer un tel guidage pédagogique. Actuellement, Marie doit organiser les activités dans des dossiers séparés, en fonction des thématiques mises en jeu. Ensuite, au fur et à mesure de l'avancement du cours magistral, elle donne accès aux étudiants à un ou plusieurs dossiers contenant les exercices pertinents. Cependant, cette tâche est coûteuse en temps et la mise à disposition des dossiers doit se faire manuellement chaque semaine. De plus, le guidage n'est pas adapté à chaque apprenant.

Marie souhaite donc créer un système d'assistance qui fournit un guidage pédagogique pour afficher la liste d'activités pertinentes à réaliser par les étudiants. Cette liste est faite en fonction de la progression de chaque étudiant au sein de la plateforme. Plus précisément, deux premières activités simples pourraient être présentées aux apprenants. Deux autres activités

seraient proposées seulement si l'étudiant termine les deux premières et ne les réussit pas du premier coup. Ce guidage pédagogique assure une bonne progression des étudiants. Le système d'assistance comportant le guidage pédagogique pourra être réutilisé et amélioré pour les années suivantes.

Résumé des contributions

Afin de répondre à notre problématique de recherche, nous avons réalisé cette thèse en deux étapes : tout d'abord l'étude d'assistances existantes au sein d'applications utilisées en contexte éducatif, puis l'exploitation et l'enrichissement des modèles et outils du projet AGATE pour les adapter au contexte éducatif.

Dans un premier temps, nous avons étudié les applications utilisées par les enseignants au sein de leurs cours ainsi que les travaux existants qui proposent des systèmes d'assistance. Nous avons ainsi identifié les caractéristiques de l'assistance, et nous les avons classifiés selon qu'elles permettent de proposer de l'assistance technique (utilisation de l'application, prise en main) ou de l'assistance pédagogique (**feedbacks pédagogiques, guidage pédagogique**). **Nous avons ajouté à cette classification** les différents **modes de déroulement d'une assistance** dans le contexte éducatif.

Dans un second temps, nous avons confronté les modèles et outils proposés précédemment dans le projet AGATE aux caractéristiques de l'assistance identifiées dans le contexte éducatif. Les limites des modèles et outils précédents nous ont amené à proposer deux contributions au langage aLDEAS et au système SEPIA pour les adapter au contexte éducatif. Cette nouvelle version de SEPIA adaptée au contexte éducatif est nommée SEPIA-edu.

La première limite concernait la complexité de définition de systèmes d'assistance variés en termes de déroulement. Que ce soit dans un contexte éducatif ou non, il est important de pouvoir définir facilement et de manière explicite plusieurs modes d'articulation entre les différents éléments d'un système d'assistance. Nous avons donc proposé un **modèle d'articulation entre les règles aLDEAS** explicitant le déroulement d'une assistance et permettant de définir des systèmes d'assistance comprenant des éléments qui se déroule de manière successive, interactive, simultanée, progressive, indépendante. Nous avons associé à ce modèle un **processus d'application automatique des contraintes des modes d'articulation aux règles** qui permet de générer et/ou modifier automatiquement les règles aLDEAS en faisant en sorte qu'elles soient toujours compatibles avec le moteur d'exécution d'assistance de SEPIA. Ce modèle et ce processus ont été implémentés dans SEPIA-edu.

La seconde limite est propre au domaine éducatif. Elle concernait la complexité à définir un guidage pédagogique proposant un parcours entre différentes activités au sein d'une application existante. Nous avons tout d'abord proposé un **modèle d'activité** permettant d'identifier les activités au sein des applications. Ensuite, un **modèle de guidage pédagogique** permet de définir différents types de guidage pédagogique (libre, séquentiel, contextuel, temporel, personnalisé). Un **patron de guidage pédagogique** a été proposé pour permettre de définir un guidage pédagogique complet qui peut combiner plusieurs types de guidage pédagogique. Enfin, un **processus de transformation d'un guidage pédagogique en règles aLDEAS** permet de générer et/ou modifier automatiquement les règles aLDEAS. Ce modèle d'activité, ce patron de guidage pédagogique et ce processus ont été implémentés dans SEPIA-edu.

Plan du manuscrit

Ce manuscrit de thèse est présenté en 5 chapitres, en complément de cette introduction et d'une conclusion qui présente notamment les perspectives de cette recherche.

Le **Chapitre 2** concerne l'état de l'art sur les systèmes d'assistance en contexte éducatif. Plus précisément, nous y présentons les types d'assistance technique et pédagogique existants (cf. section 2.1), les différents modes de déroulement d'une assistance (cf. section 0). Ensuite, nous présentons les travaux permettant la mise en place d'assistance en contexte éducatif (cf. section 2.3). Enfin, nous discutons de la pertinence de ces différents travaux par rapport à notre problématique (cf. section 2.4).

Le **Chapitre 3** présente le contexte dans lequel cette thèse se situe, et notamment le projet AGATE (cf. section 3.5). Tout d'abord, nous présentons le processus d'adjonction d'un système d'assistance sur une application-cible proposé dans le projet AGATE (cf. section 3.5.1). Ce processus est composé de deux phrases : la spécification d'une assistance, et l'exécution de l'assistance. Nous présentons ensuite le langage aLDEAS permettant de modéliser un système d'assistance en règles aLDEAS (cf. section 3.5.2) et le système SEPIA opérationnalisant les deux phrases du processus pour mettre en place des systèmes d'assistance sous forme de règles aLDEAS (cf. 3.5.3). Un exemple détaillé est ensuite donné pour montrer comment le concepteur peut définir un système d'assistance en aLDEAS avec SEPIA (cf. section 3.5.4). Enfin, nous discutons du potentiel de SEPIA en contexte éducatif (cf. section 3.6). Cette discussion nous permet d'aborder deux questions de recherches : « Comment définir le déroulement d'une assistance avec SEPIA ? » et « Comment définir un guidage pédagogique avec SEPIA ? ».

Le **Chapitre 4** répond à la première question de recherche identifiée dans le Chapitre 3. Nous y présentons notre première contribution qui explicite et facilite la définition du déroulement d'une assistance en détaillant le modèle d'articulation entre règles (cf. section 4.1), le processus d'application automatique des contraintes des modes d'articulation entre règles (cf. section 4.2) et leur implémentation dans l'extension que nous proposons de SEPIA, désormais nommé SEPIA-edu (cf. section 4.3). Un exemple détaillé est ensuite donné pour montrer comment définir un déroulement d'une assistance en aLDEAS avec SEPIA-edu (cf. section 4.4).

Le **Chapitre 5** répond à la deuxième question de recherche identifiée dans le Chapitre 3. Nous y présentons notre seconde contribution qui explicite et facilite la définition d'un guidage pédagogique en détaillant le modèle d'activité (cf. section 5.1), le modèle de guidage pédagogique (cf. section 5.2), le patron de guidage pédagogique (cf. section 5.3), le processus de transformation du guidage pédagogique en règles aLDEAS (cf. section 5.4) et leur implémentation dans SEPIA-edu (cf. section 5.5). Un exemple détaillé est ensuite donné pour montrer comment définir un guidage pédagogique en aLDEAS avec SEPIA-edu (cf. section 5.6).

Le **Chapitre 6** concerne l'évaluation de nos contributions. Nous y présentons une grille d'évaluation (cf. section 6.1) qui montre les critères selon lesquels nous souhaitons évaluer nos propositions. Ensuite, nous présentons les mises à l'essai réalisées pendant la thèse (cf. section 6.2).

Chapitre 2. État de l'art sur les systèmes d'assistance en contexte éducatif

Objectif de ce chapitre

Classifier les caractéristiques d'une assistance en contexte éducatif ; étudier et évaluer les caractéristiques des travaux visant à mettre en place de l'assistance en contexte éducatif.

Introduction de ce chapitre

Dans la première phase de cette thèse, centrée sur l'étude des assistances existantes au sein d'applications utilisées en contexte éducatif (cf. section Problématique dans le Chapitre 1), nous souhaitons répondre à trois questions « qu'est-ce qu'un système d'assistance doit fournir en contexte éducatif », « comment un système d'assistance s'exécute en contexte éducatif » et « est-ce que les systèmes existants permettent en l'état de mettre en place un tel système d'assistance ? ».

Pour répondre aux deux premières questions, nous avons établi une classification des caractéristiques de l'assistance en contexte éducatif *via* une étude sur les applications proposées et/ou utilisées en contexte éducatif. Dans cette classification, les **types d'assistance** (cf. A dans la Figure 1), où l'on distingue assistance technique (cf. T dans la Figure 1) et assistance pédagogique (cf. P dans la Figure 1), permettent de savoir ce qu'un système d'assistance doit fournir en contexte éducatif, alors que le **déroulement d'une assistance** (cf. D dans la Figure 1) permet de savoir comment s'exécute l'assistance.

Pour répondre à troisième question, nous avons fait une étude des travaux qui permettent de **mettre en place** ces types d'assistance, ainsi que le déroulement d'une assistance pour pouvoir **discuter** de leurs limites par rapport aux caractéristiques de l'assistance identifiées en contexte éducatif, ainsi que par rapport à notre approche visant à proposer un moyen unifié permettant à un enseignant de mettre en place une assistance technique et pédagogique sur des applications existantes en adoptant une approche épiphyte et générique. Ce moyen ne doit pas exiger de compétences en programmation, mais seulement des connaissances en pédagogie, pour que les enseignants concentrent leurs interventions sur le contenu pédagogique.

Contributions de ce chapitre

Classification des caractéristiques de l'assistance en contexte éducatif.

Publications liées à ce chapitre :

Thai, L.-V., Jean-Daubias, S., Lefevre, Ginon, B. : Model of articulation between aLDEAS assistance rules. DC-CSEU (2016).

Thai, L.-V., Jean-Daubias, S., Lefevre, Ginon, B. : Model of articulation between elements of a pedagogical assistance. EC-TEL (2016).

Thai, L.-V. : Concevoir un guidage pédagogique à travers un système d'assistance épiphyte. Démonstration à RJC-EIAH (2016).

Thai, L.-V., Ginon, B., Jean-Daubias, S., Lefevre, M., Pierre-Antoine, C. : Modèle d'articulation entre les règles définissant un système d'assistance aLDEAS. RJC-IA (2015).

Thai, L.-V., Ginon, B. : Exploitation d'assistances épiphytes en contexte éducatif. RJC EIAH (2014).

Ginon, B., Thai, L.-V., Jean-Daubias, S., Lefevre, M., Pierre-Antoine, C. : Adding epiphytic assistance systems in learning applications using the SEPIA system. EC-TEL (2014).

Dans le cadre de cette thèse, nous nous intéressons à la mise en place d'un système d'assistance pour aider les apprenants, qui sont les utilisateurs finaux des applications-cibles, qu'elles soient pédagogiques ou non-pédagogiques mais utilisées en contexte éducatif. Pour pouvoir créer un tel système d'assistance, il est nécessaire de savoir ce qu'un système d'assistance doit fournir aux apprenants et comment. L'identification et la classification des caractéristiques de l'assistance en contexte éducatif deviennent donc indispensables. La Figure 1 illustre une vue d'ensemble de notre classification des caractéristiques de l'assistance en contexte éducatif, classification que nous allons détailler dans les deux sections suivantes.

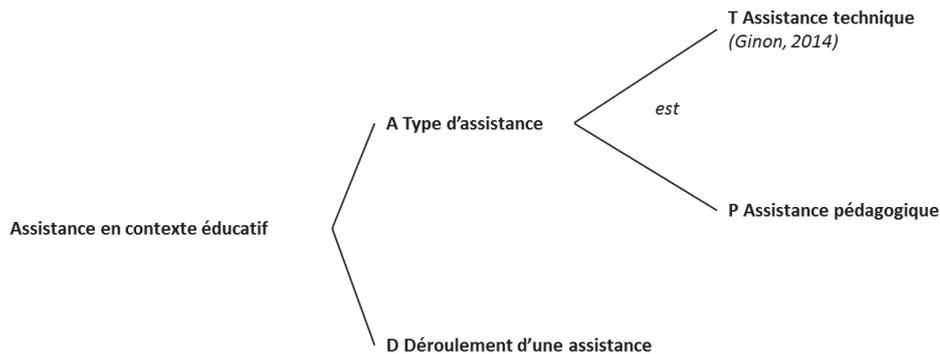


Figure 1 : Vue d'ensemble de notre classification des caractéristiques de l'assistance en contexte éducatif

Nous avons distingué les assistances existantes par leur objectif en **deux types d'assistance** (cf. A dans la Figure 1) : **assistance technique** (T) et **assistance pédagogique** (P). Ainsi, l'assistance technique aide les apprenants à pallier des difficultés de prise en main et d'utilisation des applications, alors que l'assistance pédagogique enrichit ces applications avec des compléments pédagogiques : feedbacks pédagogiques, activités pédagogiques et guidages pédagogiques.

Par ailleurs, l'assistance ne se limite pas à l'affichage de messages ou de flèches, mais elle intègre aussi la façon dont ces affichages s'articulent. Par exemple, un ensemble de messages peut être fourni de manière successive. L'utilisation combinée de plusieurs types d'assistance pédagogique rend le déroulement d'une assistance complexe. Nous avons donc également étudié dans cet état de l'art les différents **déroulements possibles d'une assistance** (cf. D dans la Figure 1) en étudiant les modes de déroulements existants.

Nous avons utilisé et étudié un ensemble d'applications (cf. Figure 164 dans l'Annexe A page 207). Nous avons utilisé certaines de ces applications, pourvues d'assistance, pour établir notre classification de l'assistance en contexte éducatif. Certaines de ces applications sont pédagogiques, d'autres non. Elles peuvent être conçues dans le cadre de recherches en EIAH, par des entreprises privées ou directement par des enseignants pour répondre à leurs besoins. La diversité des applications retenues dans notre étude a pour objectif de fournir une classification des assistances en contexte éducatif la plus complète possible.

2.1. Les types d'assistance

2.1.1 Assistance technique

L'assistance technique est proposée pour réduire les difficultés de prise en main et d'utilisation d'applications. Dans un contexte éducatif, ces difficultés peuvent compromettre l'acquisition des connaissances par les apprenants. Suite à une étude bibliographique, (Ginon, 2014) a identifié, catégorisé et hiérarchisé les besoins d'assistance technique (cf. Figure 2). Nous repartons de ce travail récent et complet pour le compléter par notre classification des assistances en contexte éducatif.

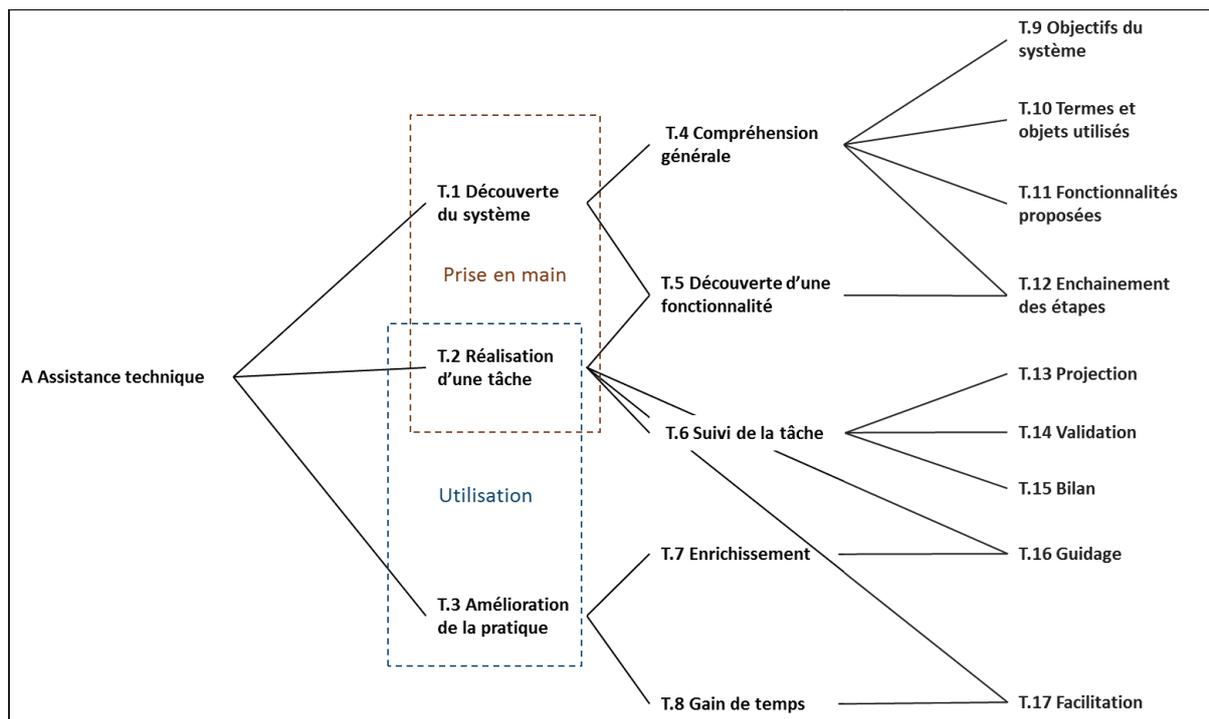


Figure 2 : Classification des caractéristiques de l'assistance technique (Ginon, 2014)

(Ginon, 2014) a identifié trois principaux besoins d'assistance (cf. T dans la Figure 2) : la découverte du système (T.1), la réalisation d'une fonctionnalité (T.2) et l'amélioration de la pratique (T.3).

Une assistance pour aider à la découverte du système (cf. T.1 dans la Figure 2) peut faciliter la compréhension du système (T.4) et / ou la découverte d'une fonctionnalité (T.5). L'utilisateur a besoin de comprendre à quoi sert le système (T.9), quels sont les termes et objets manipulés dans le système (T.10) et comment on peut les utiliser, quelles sont les fonctionnalités proposées par le système (T.11) et le cas échéant quelles sont les étapes nécessaires à la réalisation d'une tâche (T.12).

Pour aider à la réalisation d'une tâche (T.2), une assistance peut proposer la découverte d'une fonctionnalité (T.5), mais aussi le suivi de la tâche (T.6) et éventuellement la facilitation de la tâche (T.17). Ainsi, un suivi de progression de la tâche peut aider l'utilisateur à mieux comprendre ce qu'il est en train de faire dans le cas d'une tâche longue et complexe. Le suivi

peut se faire par projection (T.13), validation (T.14) ou grâce à un bilan. La projection consiste à montrer à l'utilisateur le travail qu'il a réalisé afin de l'aider à mieux comprendre ce qu'il a fait, et éventuellement à repérer ce qu'il doit rectifier pour obtenir le résultat souhaité. Cette projection peut notamment se faire sous forme d'un aperçu du travail, ou d'une instanciation du travail sur un exemple. Une validation (T.14) consiste à valider le travail de l'utilisateur pour lui confirmer que ce qu'il a fait est correct, ou au contraire à mettre en évidence les problèmes ou les incohérences de ses productions. Un bilan (T.15) consiste à résumer le travail de l'utilisateur pour qu'il visualise ce qui a été fait et ce qu'il reste à faire. Par ailleurs, la réalisation d'une tâche peut entraîner un besoin de guidage (T.16) afin notamment d'aider l'utilisateur à comprendre ce qu'il doit ou peut faire à un moment donné. Enfin, Il est possible de faciliter la tâche de l'utilisateur (T.17) en pré-réalisant cette tâche partiellement ou totalement, tout en lui permettant de modifier ou compléter les propositions qui lui sont faites.

Pour améliorer la pratique (T.3) de l'utilisateur dans la tâche, une assistance peut d'une part suggérer l'enrichissement de la pratique de l'utilisateur (T.7), notamment en mettant en évidence des options ou fonctionnalités offertes par le système et non utilisées, et d'autre part permettre un gain de temps dans la réalisation d'une tâche (T.8), par l'automatisation de certaines parties de la tâche par exemple.

L'ensemble des besoins d'assistance présentés ont été identifiés dans des domaines variés, pour des applications de types variés et concernent plusieurs types d'utilisateurs (allant de l'utilisateur novice à l'utilisateur expert). L'assistance technique à la découverte du système, à la réalisation d'une tâche et à l'amélioration de la pratique favorisent la **prise en main** et l'**utilisation** d'un système informatique. Ainsi, la découverte du système permet sa prise en main alors que l'assistance à la réalisation d'une tâche et l'amélioration de la pratique favorisent son utilisation.

2.1.2 Assistance pédagogique

L'assistance pédagogique est au cœur d'un système d'assistance dans le contexte éducatif. Contrairement à l'objectif de l'assistance technique, le but d'une assistance pédagogique n'est pas nécessairement de permettre à un apprenant de terminer sans erreur et au plus vite une activité d'apprentissage. L'objectif principal est de faciliter l'acquisition des connaissances en jeu.

Dans le contexte éducatif, les apprenants acquièrent des connaissances en réalisant des activités d'apprentissage. L'assistance pédagogique doit permettre d'ajouter ou de compléter une **activité d'apprentissage** au sein des applications-cibles. De plus, pour une activité, l'assistance pédagogique permet de fournir des **feedbacks pédagogiques** qui assistent la réalisation de cette activité par l'apprenant. Pour un ensemble d'activités, l'assistance pédagogique peut fournir un **guidage pédagogique** qui les oriente dans le choix des activités d'apprentissage à réaliser.

2.1.2.1 Assistance à l'activité d'apprentissage

Dans notre contexte, une **activité d'apprentissage** représente les tâches que doivent accomplir les apprenants sur l'application pour atteindre les objectifs pédagogiques spécifiés : acquisition de connaissances ou de compétences, évaluation, remédiation, etc. Une activité est

associée à des ressources informatiques (pages web, fenêtres du logiciel, exercices, questions, fichiers...) utilisées par les apprenants. Une activité peut être également associée à des ressources non-informatiques (fiches papier, calculatrice, etc.). Dans cette thèse, nous nous intéressons uniquement aux activités utilisant des ressources informatiques. De plus, nous nous intéressons à des activités d'apprentissage délimitées au sein des applications-cibles. Ces activités peuvent être préexistantes ou non.

La plupart de logiciels éducatifs proposent des activités d'apprentissage aux apprenants. Par exemple, ActiveMath (Melis *et al.*, 2001) propose des activités dans lesquelles les apprenants suivent un cours ou font des exercices en mathématiques. Cependant, certains logiciels ne proposent pas d'activités d'apprentissage, essentiellement les logiciels non-pédagogiques (NetBeans, Word, Excel, etc.).

Pour compléter les applications existantes, les enseignants peuvent ajouter des activités d'apprentissage qui n'existaient pas initialement dans les applications. Ainsi, plusieurs plateformes permettant de générer des exercices comme ASKER (Lefevre *et al.*, 2015), MIRTO (Antoniadis *et al.*, 2005), Exercise Maker (Yurievich, 2015). Par exemple, ASKER permet aux enseignants, *via* une interface auteur, de générer des exercices qui seront utilisées par les apprenants en auto-évaluation. Mais cette création n'est la plupart du temps pas possible dans les applications, spécialement dans le cas des applications non-pédagogiques. Ainsi, NetBeans ne permet pas la création d'activités d'apprentissage. Pour des applications de ce type, les enseignants peuvent proposer des activités d'apprentissage externes à l'application, en fournissant une vidéo ou un document explicitant ces activités.



Figure 3 : Instructions explicitant l'activité dans AMBRE-add (Nogry, Guin and Jean-Daubias, 2008)

Expliciter l'activité d'apprentissage à réaliser au sein de l'application est nécessaire pour que les apprenants sachent ce qu'ils doivent faire dans l'application. Les **instructions explicitant l'activité** peuvent être fournies de différentes manières : par les applications-cibles, par des documents qu'ils soient sous forme papier ou numérique, par des vidéos, etc. Par exemple, AMBRE-add (Nogry, Guin and Jean-Daubias, 2008) propose des messages pour expliquer l'activité (cf. Figure 3). Chaque message, tel que « Choisis un schéma pour représenter le problème » ou « Montre ce qu'on recherche », informe l'apprenant sur ce qu'il devra faire à chaque étape. Le logiciel Tri sélectif (Jean-Daubias, 2011) propose quant à lui aux apprenants

une activité sous forme d'un jeu dans lequel ils doivent mettre les déchets dans la poubelle pertinente. Des instructions explicitant cette activité sont prononcées oralement par le logiciel.

Pour réaliser les activités d'apprentissage fournies par les applications, les apprenants utilisent des ressources informatiques (un lien d'une page web, le contenu de la fenêtre de l'application, des fichiers, etc.). Pour que les apprenants puissent accéder aux ressources informatiques, l'enseignant ou l'application peuvent les **guider** en leur fournissant des instructions ou en les dirigeant directement vers ces ressources. Par exemple, Aplusix fournit les instructions sur les ressources informatiques à travers sa carte d'exercices (cf. Figure 4). Cette carte organise les exercices selon le type de nombres utilisés dans l'exercice (nombres entiers, nombres décimaux, fractions, radicaux) et selon le niveau de difficulté (allant de 1 à 10). Dans l'exemple d'ASKER, l'enseignant peut fournir aux étudiants le lien pour accéder à l'exercice à faire sur l'application.

Les activités d'apprentissage sont indispensables en contexte éducatif. Toutefois, l'activité d'apprentissage souhaitée par l'enseignant n'est pas toujours disponible dans l'application utilisée. Un système d'assistance est une solution pour permettre à l'enseignant d'ajouter et/ou de compléter une activité d'apprentissage, mais aussi pour guider l'apprenant dans son utilisation des ressources informatiques ou pour expliciter l'activité à réaliser.

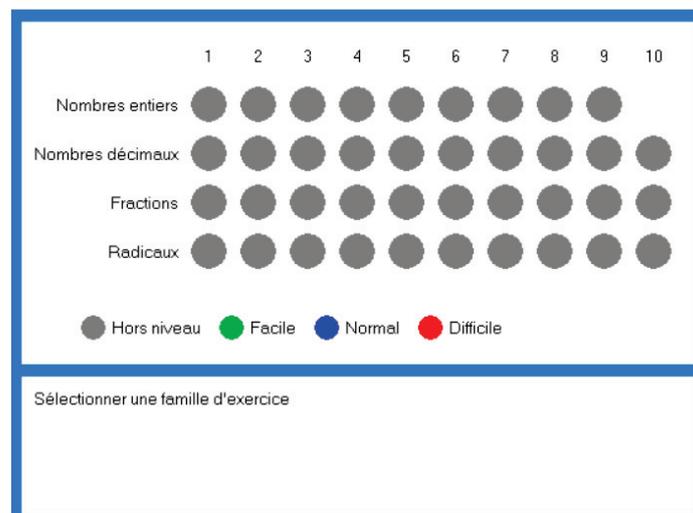


Figure 4 : Carte d'exercices sur Aplusix

2.1.2.2 Feedbacks pédagogiques

Les **feedbacks pédagogiques** assistent l'apprenant dans la réalisation d'une activité pour favoriser l'apprentissage avec les applications-cibles. Les feedbacks pédagogiques sont variés afin de s'adapter à la variété des apprenants ainsi qu'à la variété des tâches à réaliser. Ainsi, les apprenants sont différents du point de vue de leur niveau d'apprentissage, de leurs capacités physiques, de leurs capacités de compréhension, etc. De plus, pour un même apprenant, fournir le même feedback chaque fois qu'il demande de l'aide n'est pas forcément pertinent et ne va dans certains cas pas pouvoir l'aider dans ses activités d'apprentissage. Les applications pédagogiques existantes proposent donc différents types de feedbacks pédagogiques. Au travers de l'étude de nombreuses applications, nous avons identifié six types de feedbacks pédagogiques que nous détaillons dans les sections suivantes.

2.1.2.2.1 Exemples, contre-exemples

Tout d'abord, les feedbacks pédagogiques peuvent être des **exemples** aux problèmes rencontrés par les apprenants ou des **contre-exemples** donnés pour montrer la non-pertinence d'une conjecture erronée de l'apprenant.

AMBRE-add (Nogry, Guin and Jean-Daubias, 2008) fournit ainsi des exemples aux apprenants sur lesquels ceux-ci peuvent s'appuyer pour résoudre des problèmes de calcul élémentaire. Pour un modèle de problème donné, le système fournit un exemple sous forme de message (cf. Figure 5).

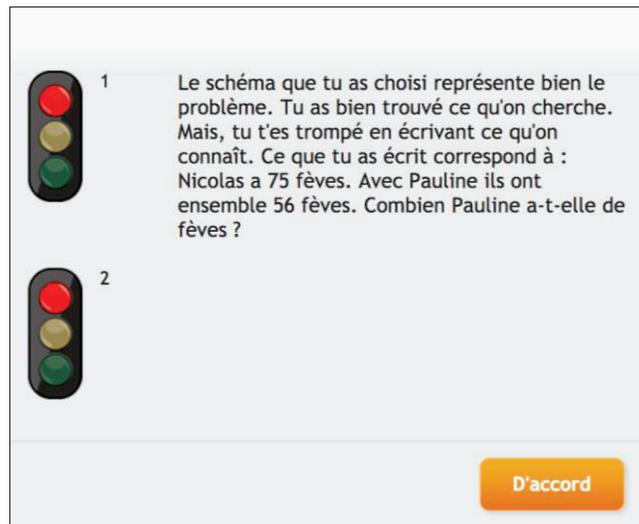


Figure 5 : Exemple proposé à l'élève dans AMBRE-add (Nogry, Guin and Jean-Daubias, 2008)

MathEnPoche (Mathenpoche, 2016) fournit quant à lui un exemple quand l'apprenant en cours de réalisation d'une question demande l'aide. L'exemple est affiché sous forme d'un message (cf. Figure 6).

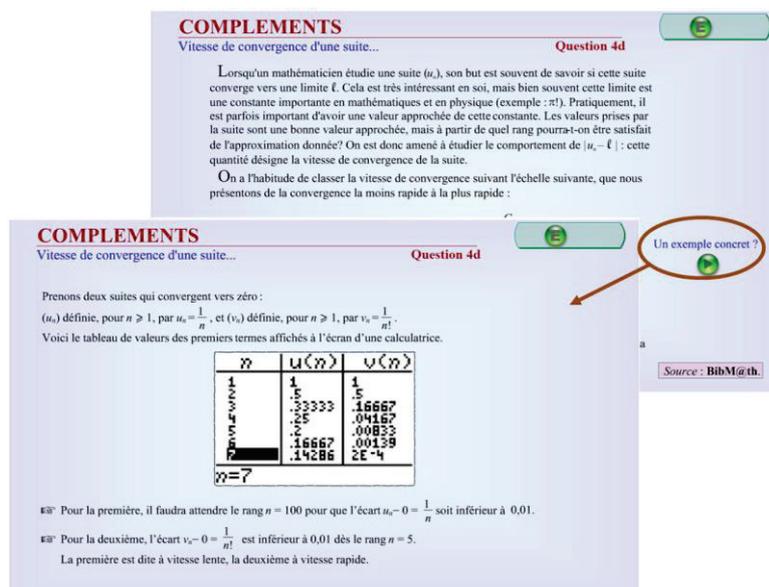


Figure 6 : Exemple proposé à l'élève dans (Mathenpoche, 2016)

2.1.2.2.2 Indices

Les feedbacks pédagogiques peuvent être des **indices** qui apportent aux apprenants une aide pour réaliser l'activité sans toutefois leur fournir la solution.

Par exemple, Aplusix (Bouhineau *et al.*, 2002) fournit aux apprenants des indices pour réaliser une étape dans la résolution d'une équation mathématique. Pour un exercice d'addition (cf. Figure 7), Aplusix donne ainsi un indice qui suggère à l'apprenant de commencer par ajouter les deux premiers nombres de l'équation. Cet indice est fourni sous forme d'un message associé à la mise en valeur des deux nombres concernés.

ActiveMath (Melis *et al.*, 2001), pour que les apprenants puissent réaliser l'exercice, peut quant à lui fournir différents indices de plus en plus détaillés. Pour un exercice de calcul de la limite d'une fonction, le premier indice (cf. 1 dans la Figure 8) est proposé à la première demande d'aide de l'apprenant. Si l'apprenant continue à demander de l'aide (cet indice ne lui a pas permis de résoudre l'exercice), ActiveMath propose un deuxième indice plus détaillé (cf. 1 dans la Figure 8).

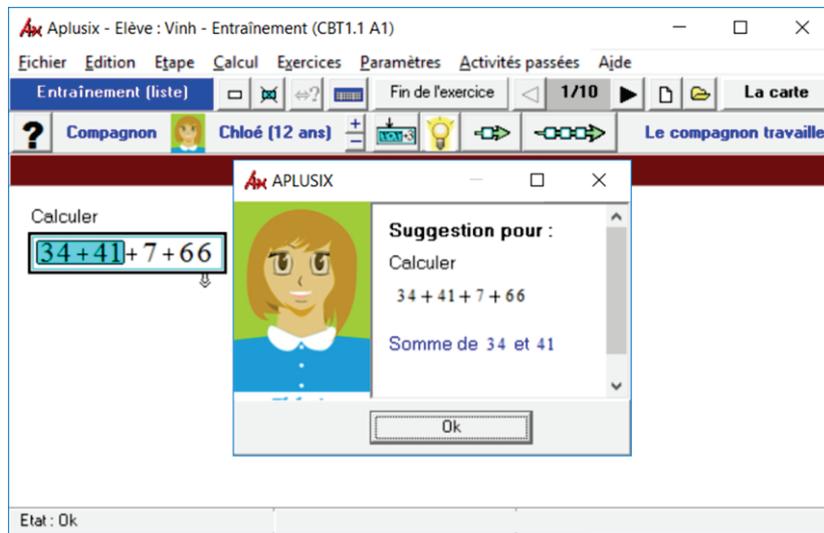


Figure 7 : Indice proposé par Aplusix

Calculating limits of trig functions 1 ★★☆☆

Calculate the following limit (input only the final result):

① The limit seems undefined (0/0) because of the tan x. Use a trigonometric formula to change it.

② Change f(x) = $\frac{\tan x}{x}$ to f(x) = $\frac{\sin x}{\cos x \cdot x}$ or f(x) = $\frac{\sin x}{x} \cdot \frac{1}{\cos x}$.
Can you find it now? Do you remember $\lim_{x \rightarrow 0} \frac{\sin x}{x}$?

③ We have
 $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$ (see why if you don't remember)
 $\lim_{x \rightarrow 0} \frac{1}{\cos x} = \frac{1}{\cos 0} = \frac{1}{1} = 1$.
 So now it should be too obvious.

$\lim_{x \rightarrow 0} \frac{\tan x}{x} =$

Activez l'éditeur d'expressions

Figure 8 : Les indices (1,2) et la solution (3) proposée par ActiveMath

2.1.2.2.3 Explications

Les feedbacks pédagogiques peuvent consister en des **explications** sur les connaissances à acquérir (cours, concepts, étapes de réalisation, etc.) ou sur les erreurs. Ces explications permettent aux apprenants de mieux comprendre les connaissances en jeu ou les erreurs qu'ils ont commises.

Par exemple, IXL Learning (IXL, 2016) fournit souvent des explications en fin d'exercice. Pour un exercice sur la parité d'un nombre entier, IXL Learning explique ainsi à l'apprenant que les nombres pairs se terminent par 0, 2, 4, 6, 8 et que les autres nombres se terminent 1, 3, 5, 7, 9 (cf. Figure 9). Le système explique que 26 est donc pair car il se termine par 6. Cette explication est représentée à l'apprenant sous forme d'un message.

Quant à lui, ActiveMath (Melis *et al.*, 2001) explique certains concepts mathématiques en fournissant leur définition à l'apprenant. Par exemple, le concept de « dérivée » est associé à un lien vers une page web (cf. Figure 10). Lors du clic sur ce lien, cette page est ouverte et affiche la définition de ce concept.

Explanation

review Is 26 even or odd?

You answered:
 odd

remember A number is even if it ends in 2, 4, 6, 8, or 0.
 A number is odd if it ends in 1, 3, 5, 7, or 9.

solve 26 ends in 6. 26 is even.

Figure 9 : Explications proposées par IXL Learning

Différentes notations pour le quotient différentiel et la dérivée

Au lieu des variables x_0 et x , on utilise souvent x_0 et h , où h représente la différence $x-x_0$. Nous avons ainsi $x = x_0+h$ et $x = x_0$ si et seulement si $h = 0$. Le quotient différentiel devient donc une fonction de h , dire $g(h)$, et on obtient pour la dérivée de f en x_0 :

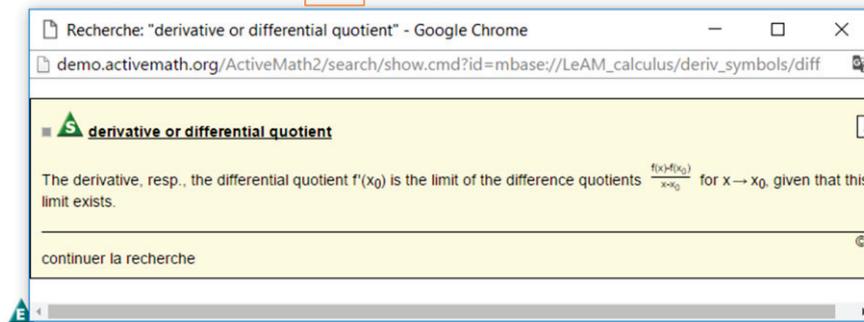


Figure 10 : Explication sur un concept proposée par ActiveMath

Les explications peuvent également jouer le rôle d'indices. La différence entre explications et indices se situe dans l'intention de leur utilisation. Ainsi, si les indices sont fournis pour que les apprenants puissent résoudre des problèmes, les explications sont fournies pour qu'ils puissent mieux comprendre les connaissances en jeu et leurs erreurs. Comme dans l'exemple de l'exercice d'IXL Learning présenté ci-dessus, où l'explication sur la parité est fournie à la fin de la réalisation pour que les apprenants apprennent à distinguer les nombres pairs des nombres impairs, mais où cette explication pourrait être considérée dans un autre contexte comme un indice pour aider l'apprenant à trouver la solution.

2.1.2.2.4 Présentation de la solution

Si les types de feedbacks précédents ne peuvent pas aider les apprenants, **la présentation de la solution** pourra les débloquent dans la réalisation de l'activité. Présenter la solution n'est pas forcément conseillé car cela ne fait pas travailler les apprenants. La présentation de seulement une partie de la solution est souvent préférable pour débloquent l'apprenant sur une étape de l'activité.

Par exemple, ActiveMath finit par fournir la solution à l'apprenant s'il demande de l'aide plusieurs fois. Ainsi, pour un exercice de calcul de la limite d'une fonction, les indices (cf. 1 et 2 dans la Figure 8) sont fournis lors des deux premières demandes d'aide. Si l'apprenant continue à demander de l'aide, la solution est donnée (cf. 3 dans la Figure 8). Cette solution est représentée sous forme d'un message.

Dans d'autres applications, la solution peut être fournie en réalisant l'activité à la place des apprenants. Par exemple, Aplusix donne une partie de la solution à l'apprenant en effectuant une étape de résolution de l'équation à sa place.

2.1.2.2.5 Diagnostic

Les feedbacks peuvent être effectués sous forme de **diagnostic** qui permet la vérification des réponses de l'apprenant pour l'activité. Grâce au diagnostic, les apprenants peuvent identifier leurs erreurs. Plusieurs applications utilisent le diagnostic comme feedback à l'apprenant (Matou Matheux, Aplusix, ASKER, etc.).

Ainsi, Matou Matheux (Matheux, 2016) propose systématiquement un diagnostic avec ses exercices de mathématiques. Dans la plupart des exercices, le diagnostic est fourni *via* un

message, par exemple « C'est juste, continue », qui notifie l'apprenant si sa réponse est correcte ou pas (cf. partie gauche dans la Figure 11). Cependant, dans d'autres exercices, Matou Matheux fournit le diagnostic par le déblocage de l'étape suivante. Par exemple, pour un exercice d'addition (cf. partie droite de la Figure 11), dès que l'apprenant a correctement écrit un nombre dans la première case jaune, les deux dernières cases apparaissent.

Le diagnostic peut également être fourni par la mise en valeur des composants concernés. Par exemple, Asker affiche en rouge les mauvaises réponses et en vert les bonnes.

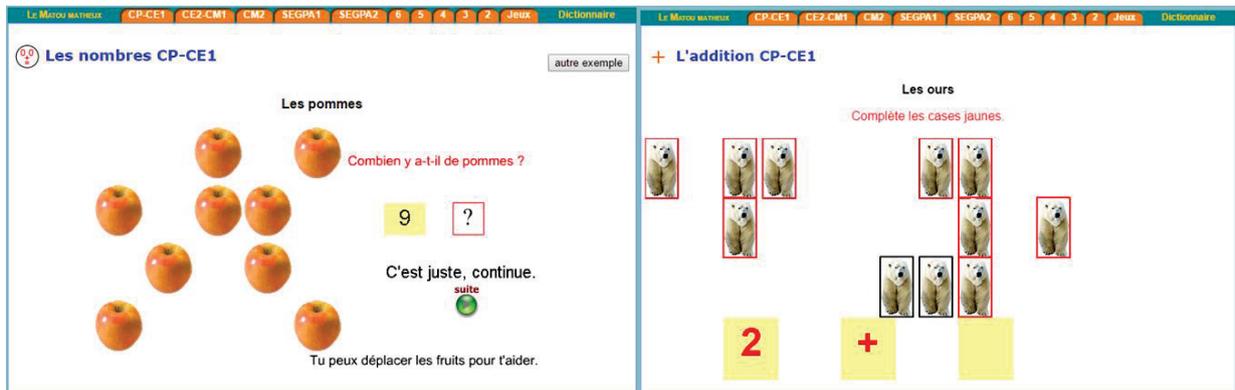


Figure 11 : Diagnostic proposé par Matou Matheux (Matheux, 2016)

2.1.2.2.6 Bilan, suivi du travail

Enfin, le feedback pédagogique peut être fait sous forme d'un **bilan** ou du **suivi du travail** de l'apprenant. Ces formes de feedback permettent de montrer à l'apprenant ce qu'il a déjà fait et ce qu'il lui reste à faire. Des détails, tel que des commentaires ou des évaluations, y sont également parfois associés pour que l'apprenant puisse savoir ce qu'il a besoin d'améliorer.

Par exemple, Educ@ffix (Ergun, 2006) fournit un bilan aux apprenants qui réalisent à distance un TP de chimie. Ce bilan représente leur progression dans les étapes du TP. La Figure 12 montre trois différents exemples du bilan proposé par Educ@ffix. L'accomplissement des étapes est représenté par une barre de progression (cf. partie gauche de la Figure 12). Ce bilan est détaillé par d'autres informations telles que les erreurs effectuées (cf. partie droite de la Figure 12), les commentaires ou les notes.



Figure 12 : Exemples de bilans proposés par Educ@ffix (Ergun, 2006)

Certaines applications fournissent un bilan ou un suivi du travail avec lequel les apprenants peuvent interagir. Par exemple, AMBRE-add affiche un suivi du travail de l'apprenant en représentant la progression des étapes de résolution d'un problème *via* des onglets actifs ou inactifs correspondant à chaque étape. Les apprenants peuvent donc savoir quelles étapes sont déjà faites et quelles étapes restent à faire.

En pratique, les différents types de feedbacks peuvent être associés et fournis ensemble. Par exemple, un message d'aide dans ActiveMath peut donner la solution ainsi que des indices. À travers les différents exemples que nous avons étudiés, nous pouvons voir que les moyens de fournir des feedbacks sont variés : affichage d'informations (message, barre de progression, pages web, etc.), mise en valeur (colorer, entourer, etc.), automatisation de tâches (remplissage automatique, clic automatique, etc.).

Nous avons identifié six types de feedbacks : exemple ou contre-exemples, indice, diagnostic, explication sur les connaissances ou les erreurs, solution, bilan ou suivi du travail. Un système d'assistance en contexte éducatif doit être capable de fournir tous ces types de feedbacks.

2.1.2.3 Guidage pédagogique

En EIAH, beaucoup de travaux portent sur la notion de « scénario » qui est parfois qualifiée de : pédagogique, d'apprentissage, d'activité ou d'encadrement. La définition d'un scénario diffère selon l'objectif de son utilisation.

Si l'on s'intéresse à la nature de l'activité dans le scénario, (Quintin, Depover and Degache, 2005) distinguent le scénario d'apprentissage du scénario d'encadrement. Les termes apprentissage et encadrement réfèrent respectivement aux activités d'apprentissage et aux activités de support (c'est-à-dire de soutien). Ainsi, dans de la conception d'une formation, les auteurs considèrent-ils le scénario pédagogique comme un ensemble structuré et cohérent constitué de deux parties :

- Le scénario d'apprentissage dont l'objectif est de décrire les activités d'apprentissage qui seront proposées et de définir leur articulation dans le dispositif pédagogique, ainsi que les productions qui sont attendues des apprenants.
- Le scénario d'encadrement dont l'objectif est de préciser le rôle des enseignants (notamment en matière de tutorat) et les modalités des interventions destinées à soutenir le scénario d'apprentissage.

Si l'on s'intéresse à la granularité du scénario, (Pernin and Lejeune, 2004) définissent le scénario d'apprentissage selon trois niveaux de granularité des unités d'apprentissage : unité de structuration pédagogique, séquence d'activités, activité élémentaire. Selon leur définition, « un scénario d'apprentissage représente la description, effectuée *a priori* ou *a posteriori*, du déroulement d'une situation d'apprentissage ou unité d'apprentissage visant l'appropriation d'un ensemble précis de connaissances, en précisant les rôles, les activités ainsi que les ressources de manipulation de connaissances, outils et services nécessaires à la mise en œuvre des activités ». Le scénario d'apprentissage décrit le déroulement :

- D'une activité élémentaire qui correspond à la granularité la plus fine de la situation d'apprentissage durant laquelle un ou plusieurs acteurs (apprenants, formateurs, tuteurs, etc.) agissent ou interagissent au sein d'un environnement défini et pour une

durée déterminée. Une activité élémentaire peut poursuivre un objectif d'apprentissage précis ou ne concourir à celui-ci que lors de son intégration au sein d'une séquence structurée. Par exemple, la situation en classe peut être suivie par un exposé oral, et pour la situation instrumentée : consulter une page web.

- D'une séquence d'activités qui correspond à une granularité moyenne de situation d'apprentissage pendant laquelle plusieurs activités élémentaires ou séquences sont organisées pour atteindre un objectif d'apprentissage déterminé en termes de connaissances ou compétences. Cette organisation doit permettre d'exprimer les relations de séquence et de parallélisme, de décrire les conditions d'enchaînement ainsi que de préciser le processus de flot des données associé. Par exemple, pour la situation en classe on peut avoir une séquence d'apprentissage de la lecture et pour la situation instrumentée on peut avoir une approche par projet comportant des activités de : répartition de rôles menés en parallèle, mise en commun des résultats et production d'une synthèse collective.
- D'une unité de structuration pédagogique qui correspond à la granularité la plus élevée de situation d'apprentissage pour laquelle un ensemble de séquences pédagogiques sont assemblées pour former une unité logique autour d'un thème d'apprentissage donné et pour un public précis. Par exemple, un cours, un module au format ECTS, une unité d'étude, un semestre, une licence, une filière, etc.

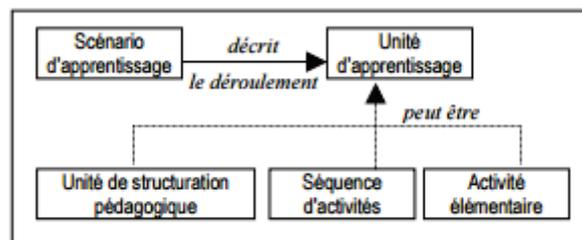


Figure 13 : Scénario et granularité des unités d'apprentissage (Pernin et Lejeune 2004)

Si l'on s'intéresse au déroulement de l'activité d'apprentissage à réaliser par les apprenants (Guéraud, 2006), (Lando, 2004) et (Hotte, Godinet and Pernin, 2007) proposent des formulations différentes. Ainsi, (Guéraud, 2006) adopte le terme de « scénario d'activité », alors que (Lando, 2004) adopte celui de « scénario pédagogique ». Mais (Hotte, Godinet and Pernin, 2007) utilisent ce même terme de « scénario pédagogique » pour décrire l'enchaînement des activités d'apprentissage.

Plus précisément, selon (Guéraud, 2006), le scénario d'activité a un triple rôle :

- il définit précisément l'activité proposée aux apprenants sur l'Objet Pédagogique Interactif ;
- il spécifie le contrôle qui sera fait de la progression de l'apprenant durant cette activité ;
- il détermine l'assistance pédagogique qui lui sera fournie automatiquement en fonction de sa progression.

Selon (Lando, 2004), un scénario pédagogique correspond au déroulement d'une activité d'apprentissage, à la définition des objectifs, à la planification des tâches, à la description des tâches des apprenants et aux modalités d'évaluation des tâches réalisées.

(Hotte, Godinet and Pernin, 2007) définissent le scénario pédagogique comme l'orchestration d'un ensemble d'activités d'apprentissage auquel s'ajoutent, d'une part, la description des ressources utiles à sa réalisation et, d'autre part, les productions de l'apprenant qui en découlent.

À partir de ces différents points de vue, nous définissons de manière concise le scénario comme permettant de décrire le déroulement d'une activité, l'enchaînement des activités et le déroulement de l'unité de structuration pédagogique de plus haut niveau incluant la description des personnes intervenant dans le scénario et leur rôle. Les activités peuvent être des activités d'apprentissage à destination des apprenants ou des activités de support aux enseignants. Dans notre contexte, nous nous intéressons plus particulièrement à l'enchaînement des activités d'apprentissage sur les applications informatiques. Le système d'assistance mis en place sur de telles applications doit permettre cet enchaînement. Pour désigner précisément un scénario dont la portée est limitée à notre contexte et pour mettre en évidence l'activité d'assistance, nous utilisons le terme « guidage pédagogique ».

Nous définissons un **guidage pédagogique** comme la préconisation d'un ensemble organisé d'activités d'apprentissage à effectuer par l'apprenant sur une ou plusieurs applications informatiques. Le guidage pédagogique met en œuvre les objectifs pédagogiques de l'enseignant. Il peut être conçu à l'identique pour l'ensemble de la classe ou de façon personnalisée pour chaque apprenant.

Cette définition du guidage pédagogique nous permet d'identifier différents types de guidages pédagogiques dans les recherches en EIAH, même si leurs auteurs adoptent d'autres termes, notamment celui de « scénario pédagogique ». Nous avons identifié, dans des travaux existants, cinq types différents de guidage pédagogique que nous présentons dans les sections suivantes.

2.1.2.3.1 Guidage libre

Dans le **guidage libre**, les activités sont librement accessibles aux apprenants. Dans plusieurs applications, les activités d'apprentissage sont toutes disponibles en parallèle et parfois organisées selon certains critères : le niveau scolaire, le thème, la catégorie, etc. Les apprenants peuvent choisir l'activité qu'ils souhaitent. Ce type de guidage libre permet au système d'assistance d'une part, de proposer un ensemble d'activités toutes pertinentes à l'apprenant parmi lesquelles il pourra faire son choix, et d'autre part, de demander à l'apprenant de réaliser plusieurs activités dont l'ordre n'est pas important.

Par exemple, le site pour l'apprentissage des mathématiques Matou Matheux (Matheux, 2016) propose plusieurs exercices depuis le niveau CP jusqu'au niveau seconde. Ces exercices sont librement accessibles pour les apprenants. Cependant, ils sont organisés selon la classe et la thématique des exercices. Par exemple, la Figure 14 montre les exercices de numération jusqu'à 10 du niveau CP-CE1. Ce site propose également des exercices qui ne sont disponibles qu'après la réussite d'autres exercices. Cela implique une séquence linéaire des exercices.

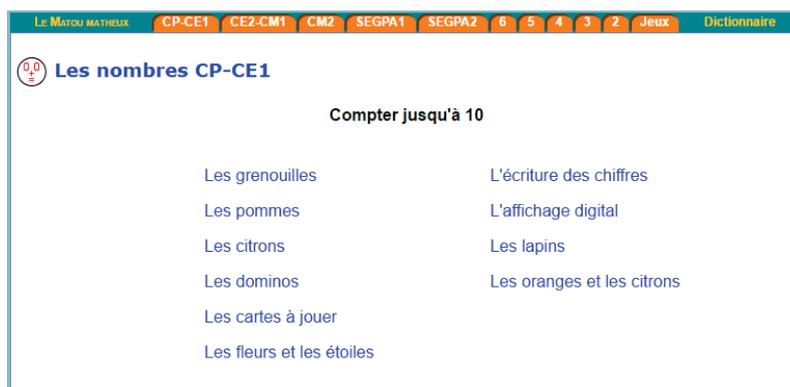


Figure 14 : Exemple d'un guidage libre sur Matou Matheux (Matheux, 2016)

2.1.2.3.2 Guidage temporel

Dans le **guidage temporel**, les activités d'apprentissage sont planifiées dans le temps. L'accessibilité ou non d'une activité dépend du moment où l'apprenant essaie d'y accéder.

Par exemple, le site de formation à distance de l'Université Lyon 1 SpiralConnect (Spiral, 2016) permet aux enseignants de créer des activités d'apprentissage pour les étudiants et de choisir un mode de publication par dates (cf. Figure 15). La visibilité du cours pour les étudiants dépend donc d'une période définie préalablement par l'enseignant.

Nom	Type	Date création	Auteur
07-02-2014 Export résultats: Cours 1 - historique	Dossier(s)	07/02/2014	stéphanie JEAN-DAUBIAS
Cours 1 - historique	Questionnaire(s)	24/09/2013	stéphanie JEAN-DAUBIAS
Cours 1 - introduction à l'IHM	Questionnaire(s)	24/09/2013	stéphanie JEAN-DAUBIAS
Passer en mode fermé	Questionnaire(s)	24/09/2013	Fabien DUCHATEAU
Passer en mode privé	Questionnaire(s)	25/11/2013	stéphanie JEAN-DAUBIAS
Passer en mode ouvert			
Passer en mode web			
Publication par dates			
Annuler			

Figure 15 : Exemple d'un type de guidage temporel sur SpiralConnect

2.1.2.3.3 Guidage contextualisé

Dans le **guidage contextualisé**, les activités sont organisées selon l'état de l'activité réalisée par l'apprenant. En fonction de l'évaluation de cette réalisation, l'apprenant est dirigé vers d'autres activités. C'est souvent le cas en remédiation quand l'apprenant rencontre des difficultés pendant la réalisation de l'activité. L'activité de remédiation permet d'aider ou d'entraîner l'apprenant pour pouvoir ensuite réussir l'activité à laquelle il avait échoué.

Le site Mathempoche (Mathempoche, 2016) propose des cours et des exercices en mathématiques. Dans l'exemple de la Figure 16, l'activité de l'apprenant consiste à répondre à une question. En cas d'erreur dans la réalisation de cette activité (cf. A dans la Figure 16), le site demande à l'apprenant de réviser la notion étudiée à travers un exemple (cf. B dans la Figure 16). Cette nouvelle activité peut être considérée comme une activité de remédiation. Si l'apprenant répond ensuite correctement à la question initiale, l'activité suivante sera ensuite proposée.

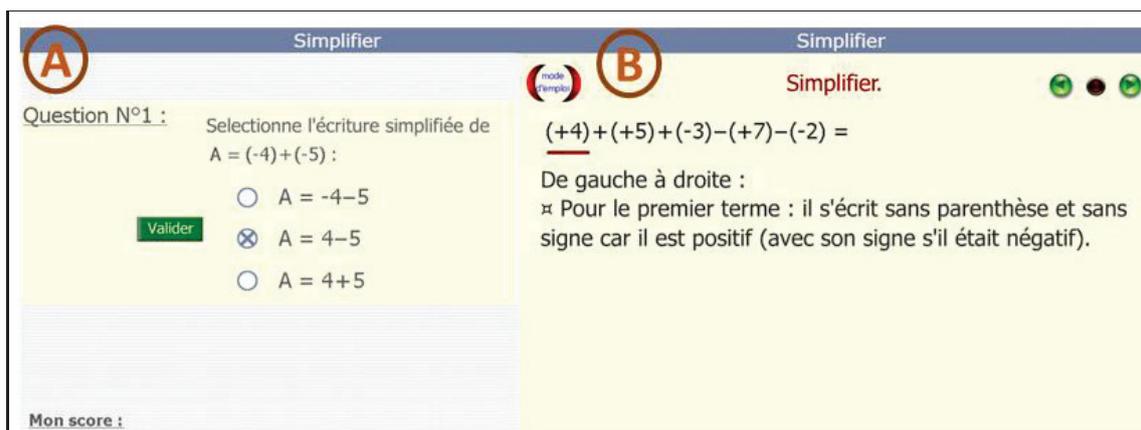


Figure 16 : Exemple d'un guidage contextualisé dans Mathenpoche (Mathenpoche, 2016)

2.1.2.3.4 Guidage séquentiel

Dans le **guidage séquentiel**, les activités sont planifiées séquentiellement et organisées selon un ordre spécifié. Une activité n'est accessible que suite à l'accomplissement de l'activité précédente. Très souvent, une activité est accomplie si l'apprenant l'a réussie. Le guidage séquentiel est donc un cas particulier du guidage contextualisé. En effet, si le guidage séquentiel prend en compte l'accomplissement de l'activité précédente pour proposer l'activité suivante, le guidage contextualisé prend en compte plusieurs états possibles de l'activité pour proposer d'autres activités.

Ce guidage pédagogique est par exemple mis en œuvre dans les sites Matou Matheux (Matheux, 2016) et Mathenpoche (Mathenpoche, 2016). L'exemple de Mathenpoche donné par la partie A de la Figure 16 montre la question que doivent faire les apprenants pour pouvoir passer à la suivante. Dans cet exemple, l'activité est la résolution d'une question, l'enchaînement des activités est donc proposé de manière séquentielle.

2.1.2.3.5 Guidage personnalisé

Dans le **guidage personnalisé**, l'organisation des activités est personnalisée pour s'adapter aux apprenants. Chaque apprenant a un profil qui peut être différent des autres. Cela implique que l'activité à réaliser doit être différente pour des apprenants de profils différents. Ainsi, un apprenant en difficulté pourra commencer par une activité facile pour prendre confiance en lui ou pour améliorer ses connaissances. En revanche, un excellent apprenant aura besoin de réaliser une activité difficile pour pouvoir se perfectionner.

Gymglish (Gymglish, 2016) propose des cours d'anglais par courriers électroniques quotidiens personnalisés. Le site prend en charge plusieurs niveaux d'anglais, de l'école élémentaire à la maîtrise professionnelle internationale. Les apprenants commencent par une phase d'évaluation sous forme de 10 leçons (cf. Figure 17) qui permet au site tester leur niveau. En fonction des résultats, le site propose ensuite des cours adaptés à chaque apprenant.

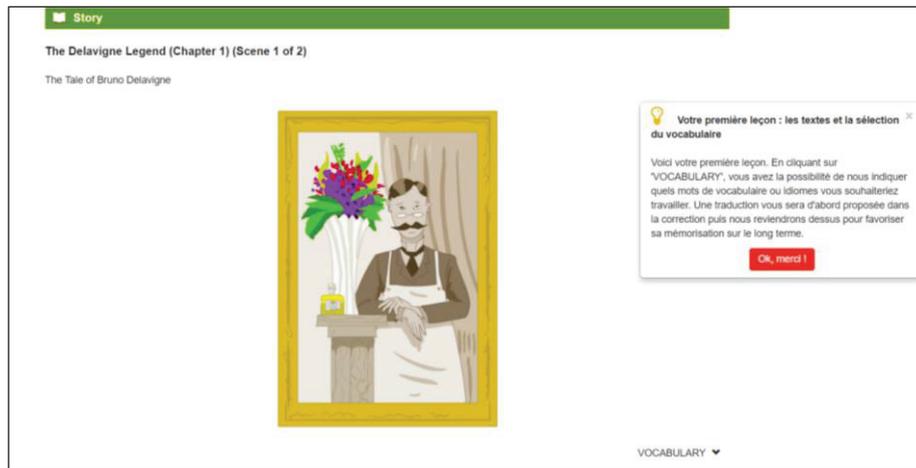


Figure 17 : La première leçon de la phase d'évaluation de Gymglish (Gymglish, 2016)

2.1.2.4 Classification des caractéristiques de l'assistance pédagogique

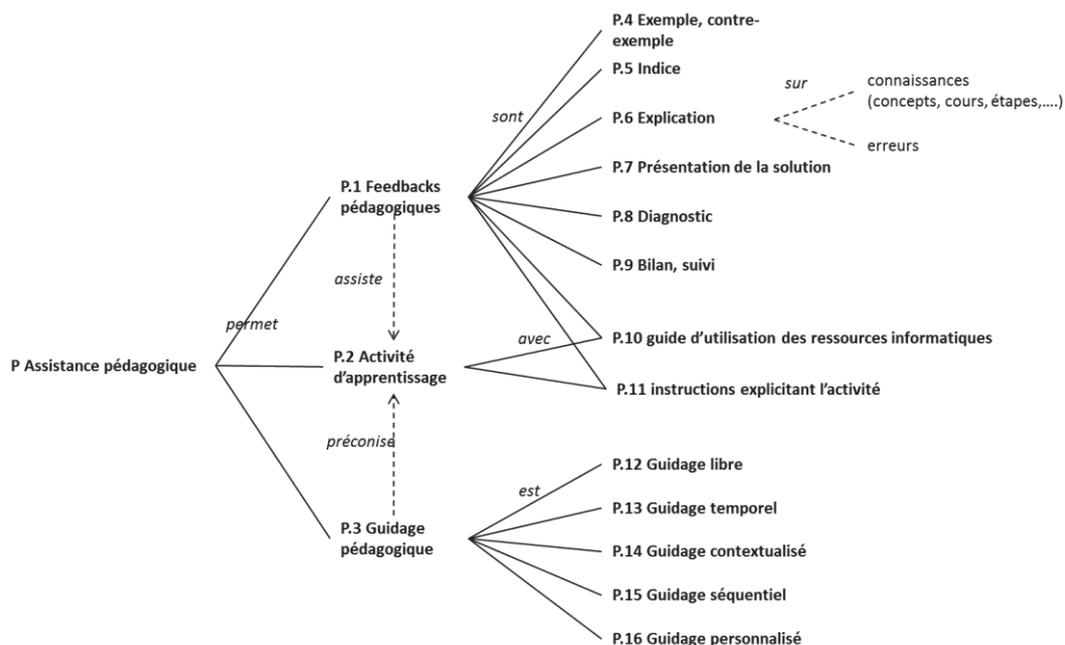


Figure 18 : Classification des caractéristiques de l'assistance pédagogique

L'assistance pédagogique étudiée dans cette partie de l'état de l'art nous a permis de proposer une classification des caractéristiques de l'assistance pédagogique en contexte éducatif. Cette classification est proposée dans la Figure 18. Elle met en évidence deux types d'assistance pédagogique : le feedback et le guidage, tous deux s'appuyant sur des activités pédagogiques. Le feedback pédagogique peut prendre la forme d'exemples ou contre-exemples, d'indices donnés à l'apprenant, d'explications, de présentation de la solution, de diagnostic, de bilan, de guide technique ou d'instructions concernant l'activité à réaliser. Quant au guidage, il peut être de type libre, temporel, contextualisé, séquentiel ou personnalisé.

2.2. Les modes de déroulement d'une assistance

Nous désignons par le terme de « **déroulement d'une assistance** » l'articulation entre les différents éléments qui constituent une assistance. Un élément peut être par exemple un message ou une mise en valeur. Les éléments peuvent s'enchaîner selon différents modes. Par exemple, des messages d'aide peuvent être affichés les uns après les autres ou simultanément. Nous avons identifié des régularités dans le déroulement des assistances que nous avons étudiées. Nous présentons dans cette section, en les illustrant d'exemples qui ne se limitent pas au contexte éducatif, les différents **modes d'articulation** entre éléments d'une assistance que cette étude nous a permis d'identifier.

2.2.1 Mode indépendant

Dans certaines applications, les éléments d'assistance sont fournis indépendamment. Il n'y a donc pas de dépendance entre eux. Chaque élément d'assistance est contraint par ses propres événements. Quand ces événements ont lieu, l'élément d'assistance concerné est lancé. Par exemple, la bulle d'aide apparaît quand l'utilisateur passe la souris sur un composant donné de l'application. Nous qualifions ce mode d'articulation de **mode indépendant**.

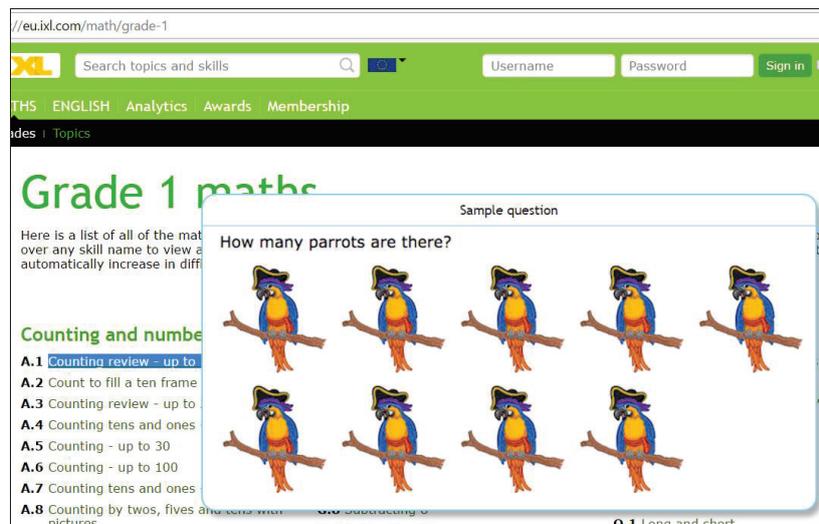


Figure 19 : Exemple de l'affichage de la vue d'ensemble d'un cours d'IXL Learning (IXL, 2016)

Ainsi, IXL Learning (IXL, 2016) présente à l'apprenant les différents cours proposés dans l'application grâce à une séquence de messages indépendants affichés dans des fenêtres pop-up. Quand l'apprenant passe la souris sur un cours, la description de ce cours est affichée. Par exemple, le cours « Counting review - up to 10 » est présenté au travers d'un exemple de question (cf. Figure 19). Quant à ActiveMath, il affiche des explications sur les concepts mathématiques au travers de pop-ups ou de pages web.

2.2.2 Mode successif

Dans certaines applications, l'assistance est fournie étape par étape, c'est-à-dire qu'un élément suit un autre élément. De telles assistances se trouvent fréquemment dans les tutoriels qui guident les utilisateurs dans l'apprentissage de l'utilisation d'une application. Nous qualifions ce mode d'articulation de **mode successif**.

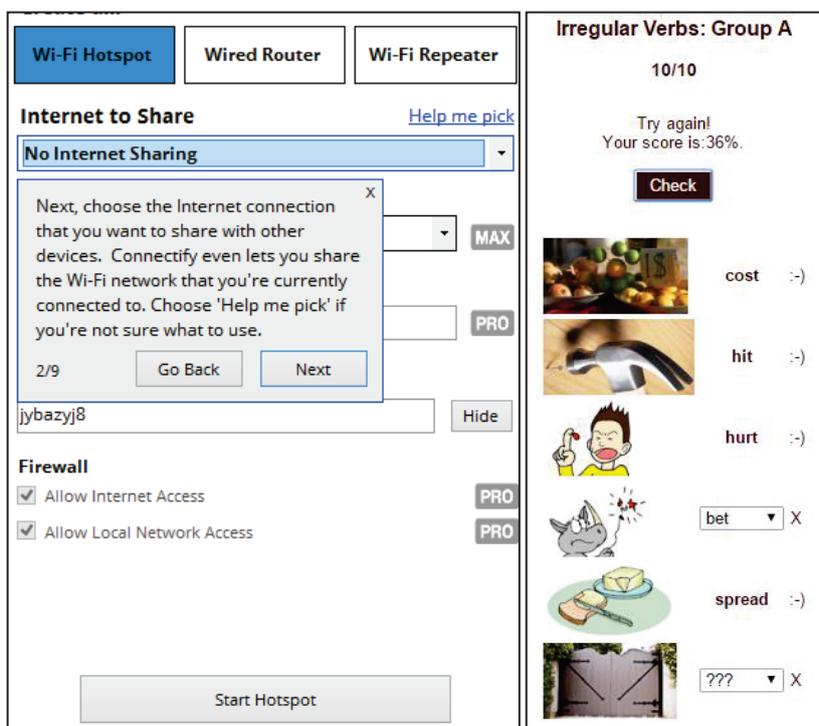


Figure 20 : Exemple d'une étape d'assistance dans le tutoriel intégré à Connectify (Connectify, 2016) et d'une étape de vérification des réponses dans EOLF (EOLF, 2016)

Par exemple, l'application Connectify (cf. partie gauche de la Figure 20) permet de créer un hotspot pour partager un accès internet. Elle fournit des messages les uns après les autres pour guider l'utilisateur dans l'utilisation de l'application. De tels messages peuvent également servir à expliquer pas-à-pas les erreurs de l'utilisateur.

2.2.3 Mode simultané

Dans les formulaires, fréquemment proposés sur le web pour la saisie des données, le diagnostic de différentes entrées de l'utilisateur peut être affiché en même temps, ce qui donne une vue complète sur l'information. L'utilisateur peut librement percevoir l'élément d'assistance dont il a besoin. Nous qualifions ce mode d'articulation de **mode simultané**.

Dans le contexte éducation, nous avons, par exemple, Hot Potatoes (Winke and MacGregor, 2001) qui permet aux enseignants de créer différents types d'exercices sous forme des pages web qui comporte un diagnostic sur les réponses des apprenants. Le diagnostic est fourni sur plusieurs réponses en même temps. Plus précisément, un exercice créé par EOLF avec Hot Potatoes à l'Université de Franche-Comté (cf. partie droite de la Figure 20) affiche l'indication « correct » ou « incorrect » simultanément pour l'ensemble des réponses.

2.2.4 Mode progressif

L'assistance peut être fournie de façon progressive. L'assistance est modifiée selon le nombre de fois que l'utilisateur se retrouve dans une même situation problématique. En particulier, l'assistance est de plus en plus importante (plus détaillée, plus précise) pour répondre à un besoin d'assistance qui se répète. Nous qualifions ce mode d'articulation de **mode progressif**.

Ainsi, dans ActiveMath (Melis *et al.*, 2001), pour résoudre un exercice, l'apprenant peut demander de l'aide de façon progressive (cf. Figure 8). Lors de la première demande, l'assistance est donnée sous forme d'un indice (1), puis sous forme d'un indice plus détaillé (2) lors d'un nouvel appel à l'aide. Lors de la troisième demande, la solution (3) est fournie pour débloquer l'apprenant. L'assistance est de plus en plus détaillée, plus concrète.

2.2.5 Mode interactif

Certaines applications, pour fournir une assistance plus pertinente, exploitent différentes sources d'informations telles que l'état de l'application, le profil de l'utilisateur, etc. L'assistance fournie est donc personnalisée et contextualisée pour pouvoir mieux assister les utilisateurs. Nous qualifions ce mode d'articulation de **mode interactif**.

The screenshot shows a window titled "Calculating limits of trig functions 1 ★★★". It contains three distinct feedback panels for the problem: $\lim_{x \rightarrow 0} \frac{\tan x}{x}$.

- Panel 1 (labeled ①):** Shows the user's input $\lim_{x \rightarrow 0} \frac{\tan x}{x} =$ followed by a red box containing "syntax error". Below it, a message says: "No this is not correct. Perhaps you did something wrong. Try again or ask for some hints to guide you through the solution."
- Panel 2 (labeled ②):** Shows the user's input $\lim_{x \rightarrow 0} \frac{\tan x}{x} = 12$ followed by a red box containing "12". Below it, a message says: "No this is not correct. Perhaps you did something wrong. Try again or ask for some hints to guide you through the solution."
- Panel 3 (labeled ③):** Shows the correct solution. It starts with "Correct. Here is a full solution:". It defines $f(x) = \frac{\tan x}{x} = \frac{\sin x}{\cos x \cdot x}$ or $f(x) = \frac{\sin x}{x} \cdot \frac{1}{\cos x}$. It then states: "But $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$ and $\lim_{x \rightarrow 0} \frac{1}{\cos x} = \frac{1}{\cos 0} = \frac{1}{1} = 1$. So, using the arithmetic of limits we have $\lim_{x \rightarrow 0} \frac{\tan x}{x} = 1 \cdot 1 = 1$."

Figure 21 : Exemples de vérification de la réponse sur ActiveMath (Melis *et al.*, 2001)

Ainsi, ActiveMath (Melis *et al.*, 2001) et Hot Potatoes (Winke and MacGregor, 2001) consultent l'état de l'application, par exemple le texte écrit par l'apprenant. Ainsi ActiveMath vérifie les réponses de l'apprenant et affiche le résultat. La Figure 21 montre trois diagnostics différents : erreur de syntaxe, réponse incorrecte et réponse correcte. Ce diagnostic s'appuie sur la consultation de la réponse de l'apprenant.

2.2.6 Classification des modes de déroulement d'une assistance

Les modes de déroulement d'une assistance étudiés dans cette partie de l'état de l'art nous permettent de proposer une classification des modes de déroulement d'une assistance en contexte éducatif. Cette classification est proposée dans la Figure 22. Nous avons identifié cinq modes de déroulement d'une assistance : indépendant, successif, simultané, progressif et interactif.

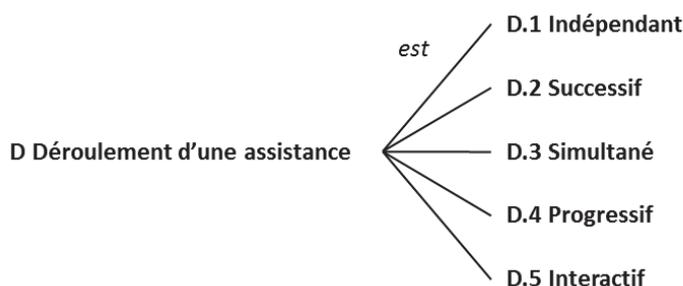


Figure 22 : Classification des modes de déroulement d'une assistance

2.3. Mise en place d'assistance en contexte éducatif

Dans cette section, nous nous intéressons aux travaux qui permettent de mettre en place une assistance, principalement en contexte éducatif. L'assistance peut être proposée par un système conseiller, par des tuteurs humains ou artificiels, au sein d'un scénario pédagogique ou incluse dans un hypermédia adaptatif. L'assistance peut également exister comme fonctionnalité de l'application.

Concernant le déroulement d'une assistance, les travaux qui permettent de définir des assistances permettent également de définir les modes de son déroulement. Cependant, les modes de déroulement d'une assistance restent implicites, car ce ne sont pas des éléments clés dans ces travaux. Nous avons donc en complément étudié Grafcet qui se focalise sur le déroulement d'un système, même si le système en question n'est pas forcément un système d'assistance.

Nous avons ensuite confronté ces travaux à notre approche pour la mise en place d'assistance (qui est épiphyte, générique et n'exige pas de compétences de programmation) et aux caractéristiques identifiées de l'assistance en contexte éducatif. Pour cette confrontation, la présentation de chaque système étudié est suivie d'un tableau positionnant ce système par rapport aux caractéristiques suivantes : pas de compétences de programmation nécessaires, approche épiphyte, approche générique, assistance technique, feedbacks pédagogiques, guidage pédagogique, déroulement d'une assistance. Le résultat de cette confrontation est exprimé visuellement selon trois niveaux : permis ✓, permis mais limité ✓ et non permis ✗.

2.3.1 Systèmes conseillers

Un système conseiller est un système qui propose une aide active à l'utilisateur d'un logiciel particulier (Winkels, 1992). Cette aide est fondée sur une analyse des actions et des productions de l'utilisateur (Paquette and Tchounikine, 2002). Parmi tous les systèmes conseillers, le système conseiller générique Marco et le système Epitalk nous intéressent particulièrement de par leur approche générique et épiphyte.

2.3.1.1 Epitalk

Le système Epitalk (Paquette, Pachet and Giroux, 1994) permet la mise en place de systèmes conseillers épiphytes sur les applications-cibles développées en Smalltalk-80 en adoptant une approche générique. Le système conseiller produit des conseils à partir de l'observation de l'interaction entre un utilisateur et une application-cible.

L'architecture d'un système conseiller dans Epitalk s'appuie sur un graphe de tâches dans lequel une tâche réalisée par l'utilisateur dans les applications-cibles est décomposée selon plusieurs niveaux hiérarchiques jusqu'aux tâches terminales. Ce graphe permet d'identifier la tâche courante de l'utilisateur pour pouvoir fournir un conseil pertinent à l'utilisateur. Le conseil est spécifié par le concepteur d'assistance avec les métadonnées qui y sont associées. Ainsi, pour un conseil, le concepteur peut spécifier un niveau d'importance qui permet au système conseiller de choisir le conseil le plus pertinent si plusieurs conseils peuvent être fournis en même temps. Les conseils peuvent également être associés à un nombre maximal d'utilisations avant leur désactivation temporaire. Le concepteur peut également déterminer si un conseil sera donné à l'initiative de l'utilisateur final ou à l'initiative du système conseiller.

Les conseils sont affichés par les agents épiphytes qui sont organisés dans un graphe d'agents épiphytes. Ce graphe est généré automatiquement et est isomorphe au graphe de tâches. Un agent est associé à chaque tâche. Les deux fonctions principales d'un agent épiphyte sont : la réception des événements sur l'application-cible envoyés par les « espions » et l'affichage des conseils en exécutant les règles, respectant le formalisme NéOpus (Pachet, 1995), associées à cet agent.

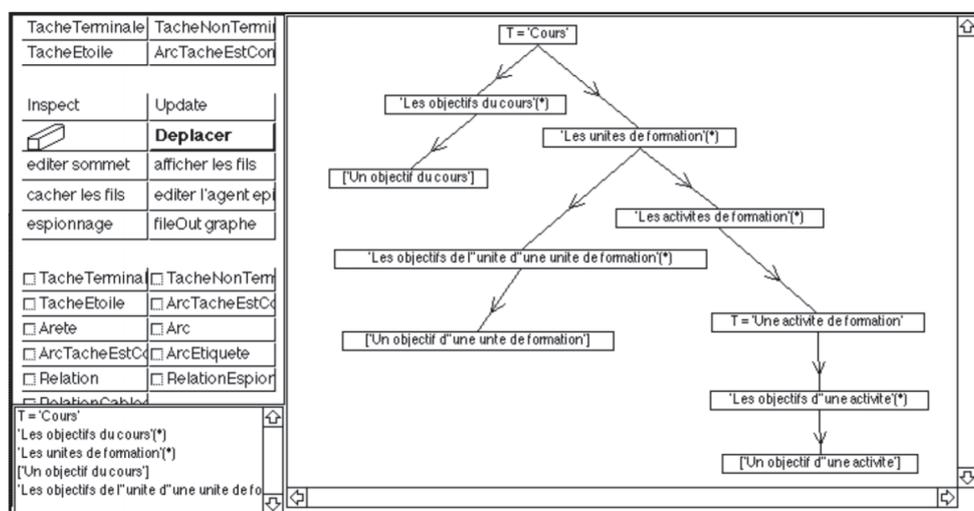


Figure 23 : Édition d'un graphe de tâches dans Epitalk (Paquette, Pachet and Giroux, 1994)

La Figure 24 confronte les spécificités du système conseiller Epitalk à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

Epitalk adopte une approche épiphyte, mais il n'est pas complètement générique. En effet, il est spécifique aux applications Smalltalk-80. De plus, des connaissances en programmation en Smalltalk-80 sont requises pour le concepteur de l'assistance, même si Epitalk limite l'usage de la programmation. Cela rend donc Epitalk très peu accessible pour des enseignants non informaticiens.

Epitalk permet de spécifier et d'exécuter de l'assistance et de fournir des feedbacks pédagogiques sur les applications-cibles de façon épiphyte, sous forme de conseils fournis aux utilisateurs. Les enseignants peuvent l'utiliser pour proposer un guidage pédagogique limité sur les applications Smalltalk-80 en spécifiant les conseils sur le choix de l'activité et les conditions de passage entre les activités dans les règles associées à chaque agent.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
Épitalk	✓	✓	✓	✓	✓	✓	✓

Figure 24 : Epitalk confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.1.2 Système conseiller générique Marco

(Richard, 2008) propose une approche permettant la mise en œuvre de systèmes conseillers pour des pages Web existantes en adoptant une approche épiphyte. De tels systèmes conseillers ont pour objectif de fournir aux utilisateurs des conseils pour la navigation au sein des pages d'un site Web existant (appelées pages-cibles). Ces conseils sont déclenchés par les utilisateurs lors des clics sur les liens. Ces conseils sont fournis soit *via* les pages de conseils greffées sur des pages-cibles par le système conseiller, soit *via* de nouvelles pages créées ou transformées à partir des pages-cibles par le système conseiller. Les premières sont greffées sous forme d'une fenêtre pop-up sur les pages-cibles. Les secondes sont ouvertes sous-forme de pages indépendantes.

Cette approche exploite un modèle de tâches. Ce modèle est un sous-ensemble du graphe du site web qui est composé : de nœuds représentant des liens et d'arcs représentant des passages entre ces liens. Ce modèle permet de modéliser les actions de navigation de l'utilisateur : les clics sur des liens par l'utilisateur. Pour pouvoir déclencher un conseil lors des clics, un modèle d'utilisation doit être élaboré en associant les conseils à des arcs identifiés dans le modèle de tâches.

En s'appuyant sur cette approche, l'architecture d'un système conseiller générique nommé Marco a été proposée. Pour pouvoir mettre en place un système conseiller sur un site web, il faut l'instancier avec le modèle de tâches du site, les conseils et le modèle d'utilisation.

Cette approche a été validée par l'exploitation de Marco pour deux sites : EducaSource et Apache. La Figure 25 montre l'exemple d'un conseil sous forme d'une fenêtre pop-up sur une page web du site EducaSource (EducaSource, 2017).



Figure 25 : Exemple d'un conseil sous forme d'une fenêtre pour EducaSource avec Marco (Richard, 2008)

La Figure 26 confronte les spécificités du système conseiller générique Marco à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

Le système conseiller générique Marco est un système épiphyte et générique. Ainsi, ce système peut être instancié pour créer des systèmes conseillers spécifiques à différentes applications web existantes. Cependant, Marco n'est pas complètement générique, car il est spécifique à un type d'applications (les sites Web). De plus, ce système est spécifique à l'assistance à l'utilisateur lors de la navigation dans un site Web : passage d'une page à une autre page par un clic sur un lien.

Il n'est donc pas possible de fournir des conseils selon des interactions variées sur des pages web. Marco permet de proposer de l'assistance technique, plus précisément de l'assistance à la découverte du système et à l'amélioration de la pratique, mais cette assistance est limitée : par exemple, elle ne permet pas d'automatiser les actions sur les pages web. Les enseignants peuvent l'utiliser pour créer une assistance, mais elle sera très limitée. Si une activité est associée à la page d'un site-cible, les enseignants peuvent créer un guidage sur l'enchaînement des pages web et créer des feedbacks fournis à chaque clic sur un lien.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
Marco	✓	✓	✓	✓	✓	✓	✓

Figure 26 : Marco confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.2 Outils auteurs

Plusieurs outils auteurs ont été proposés en contexte éducatif (Murray, 1999). Ils permettent aux enseignants, sans aucune programmation, de créer des applications, des exercices, des contenus du cours, etc. Par exemple, ASKER (Lefevre *et al.*, 2015) permet de créer des exercices proposés au sein de la plateforme, alors que Hot Potatoes permet de créer des exercices sous forme de pages HTML qui peuvent être exploitées de manière indépendante.

2.3.2.1 Hot Potatoes

Hot Potatoes (Winke and MacGregor, 2001) est un logiciel permettant à des enseignants de créer différents types d'exercices ainsi que les feedbacks pédagogiques associés. Ces exercices et feedbacks peuvent être générés sous forme de pages web qui deviennent une application indépendante. Hot Potatoes est très utilisé pour créer des pages web éducatives telles que les pages web proposé par EOLF (EOLF, 2016).

Hot Potatoes comporte actuellement cinq modules correspondant aux types d'exercices suivants : JQuiz, JMix, JCross JMax, JCloze. JQuiz permet de créer des questionnaires à choix multiples ou des questionnaires à réponses ouvertes courtes. JMix permet de créer un exercice où l'apprenant doit remettre en ordre les parties d'une phrase. JCross permet de créer des mots-croisés de façon simplifiée. JMax permet de créer des exercices de mise en correspondance où l'apprenant doit associer deux parties d'une phrase ou une image et un texte. JCloze permet de créer des exercices à trous.

Pour permettre la création des exercices, Hot Potatoes fournit des interfaces à destination des enseignants. Ainsi, la Figure 27 montre l'interface dédiée à la création d'un QCM. Les enseignants non informaticiens peuvent remplir facilement les champs correspondant aux questions et aux feedbacks à associer aux différentes réponses des apprenants.

Les feedbacks créés sont simples : il s'agit de messages qui contiennent le texte prévu par l'enseignant, ainsi qu'un symbole (« : » ou « X ») qui représente le diagnostic de la réponse de l'apprenant. Un exemple de diagnostic fourni sur les pages web de EOLF est donné en Figure 20.

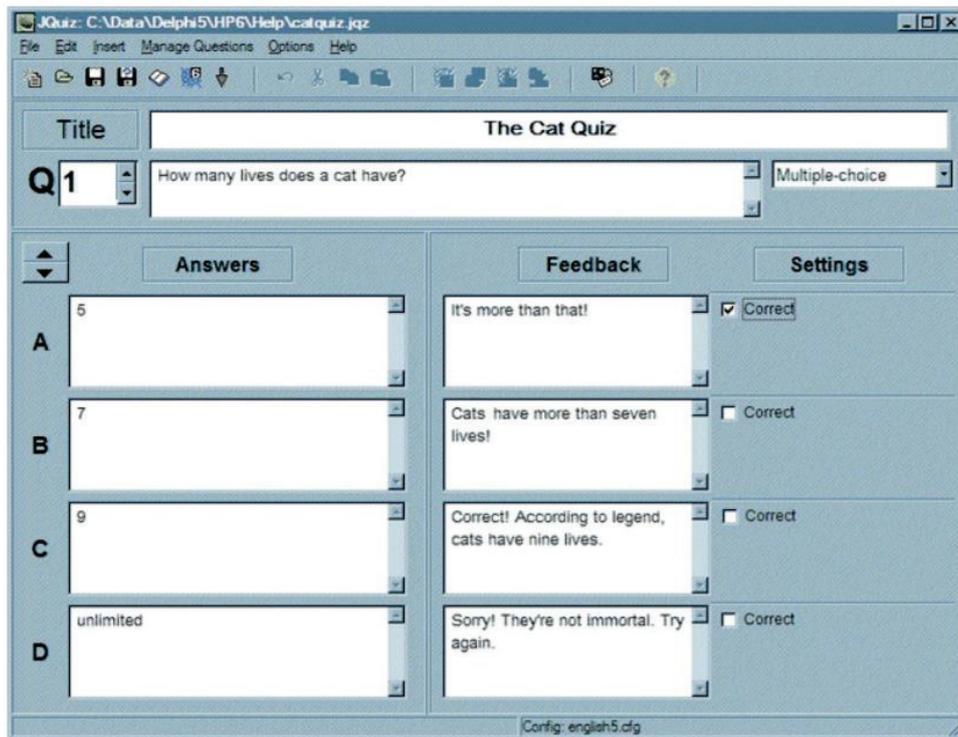


Figure 27 : Exemple de création d'un QCM avec Hot Potatoes (Winke and MacGregor, 2001)

La Figure 28 confronte les spécificités de l'outil auteur Hot Potatoes à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

Hot Potatoes permet de créer l'assistance en même temps que les exercices sans exiger de compétences de programmation. Il ne permet pas de créer de l'assistance sur d'autres applications que celles créées par Hot Potatoes. Hot Potatoes n'adopte donc ni une approche générique ni une approche épiphyte.

Hot Potatoes propose très peu de moyens pour afficher des feedbacks à l'apprenant (message et deux types de symboles). Les feedbacks sont relativement figés car ils sont associés préalablement aux réponses de l'apprenant. Hot Potatoes permet de fournir ni assistance technique, ni guidage pédagogique.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
Hot Potatoes	✓	✗	✗	✗	✓	✗	✓

Figure 28 : Hot Potatoes confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.3 Tutorat

(Gounon, Dubourg and Leroux, 2004) ont défini l'activité de tutorat comme l'accompagnement des apprenants réalisant une activité d'apprentissage. Le tutorat peut être proposé par un tuteur humain, un co-apprenant ou un dispositif informatique.

Dans le cadre de cette thèse, nous nous intéressons au tutorat fournit *via* un dispositif informatique. De nombreux travaux sur les tuteurs intelligents vont dans ce sens. Selon (Murray, 1999), un système tuteur intelligent est un système d'enseignement assisté par ordinateur reposant sur plusieurs modèles, notamment le modèle du contenu pédagogique qui spécifie ce qui doit être enseigné, et le modèle pédagogique qui spécifie les stratégies d'enseignement indiquant comment enseigner le contenu en fonction des caractéristiques des apprenants (Ohlsson, 1986; Etienne, 1987).

Par exemple, SQLT-Web (Mitrovic, 2003) est un tuteur intelligent permettant aux apprenants de créer des requêtes SQL. Il possède un module pédagogique qui fournit des feedbacks pédagogiques. Ce système est spécifique au système SQLT-Web et aux requêtes SQL. W-PAL (McNamara *et al.*, 2014) est un autre tuteur intelligent qui fournit aux élèves les instructions de stratégie pour les activités de rédaction. Il propose des feedbacks pédagogiques sur la rédaction des élèves.

L'activité de tutorat de ces systèmes est proposée au sein de chaque système et pour un domaine spécifique (SQL, la rédaction, etc.). De plus, ces systèmes n'adoptent pas une approche auteur. Ils ne sont donc pas pertinents pour mettre en place de l'assistance sur des applications-cibles très variées. Nous allons à présent présenter plus en détail deux approches qui se rapprochent des objectifs de cette thèse.

2.3.3.1 Lightweight Tutoring Agents

(Ritter and Koedinger, 1995) proposent un ensemble de standards pour concevoir des tuteurs qui communiquent avec un logiciel complexe (pédagogique ou non pédagogique) sous forme d'un modèle d'interaction entre ce logiciel et les tuteurs. Cependant, aucun outil qui permette de créer de tels tuteurs n'a été créé. Ces tuteurs sont des éléments complémentaires à des logiciels pour accompagner l'apprentissage en fournissant des conseils aux apprenants.

L'ensemble de standards est proposé grâce à la généralisation de deux exemples d'assistance faits sur Geometer's sketchpad et Microsoft Excel, deux applications issues de deux domaines différents. Geometer's sketchpad est un logiciel commercial permettant de créer des constructions géométriques. Il est utilisé à la fois en contexte éducatif et en recherche. Microsoft Excel est quant à lui un tableur utilisé dans des contextes très variés. Deux tuteurs ont été créés et intégrés à ces deux applications pour aider les apprenants à réaliser des exercices de géométrie et d'algèbre. La Figure 29 montre par exemple un indice proposé par le tuteur intégré à Geometer's sketchpad.

En analysant les similarités et les différences entre les deux exemples précédents, Ritter et Koedinger ont proposé un ensemble de standards avec quatre objets principaux : Tool, Control Center, Tutoring Agent, Curriculum Agent (cf. Figure 30). Tool est l'application à laquelle on souhaite intégrer le tuteur. Les composants de cette application doivent être suffisamment accessibles. Control Center est chargé de surveiller les actions de l'utilisateur. L'objectif de cette

surveillance est de déterminer l'accomplissement correct ou incorrect d'une tâche de l'utilisateur et éventuellement de fournir les conseils. Curriculum Agent est chargé de décider quelle tâche présenter aux apprenants. Cette décision peut être simple avec un choix d'une tâche dans une liste, ou personnalisée. Control Center est chargé de la communication entre Tutoring Agent et Tool. S'il existe plusieurs Tools et plusieurs Tutoring Agents, Control Center doit permettre le routage correct des messages entre eux. Control Center doit donc être adapté en fonction de chaque logiciel. L'existence de cet objet permet que la conception de Tool et Tutoring Agent soit indépendante l'une de l'autre.

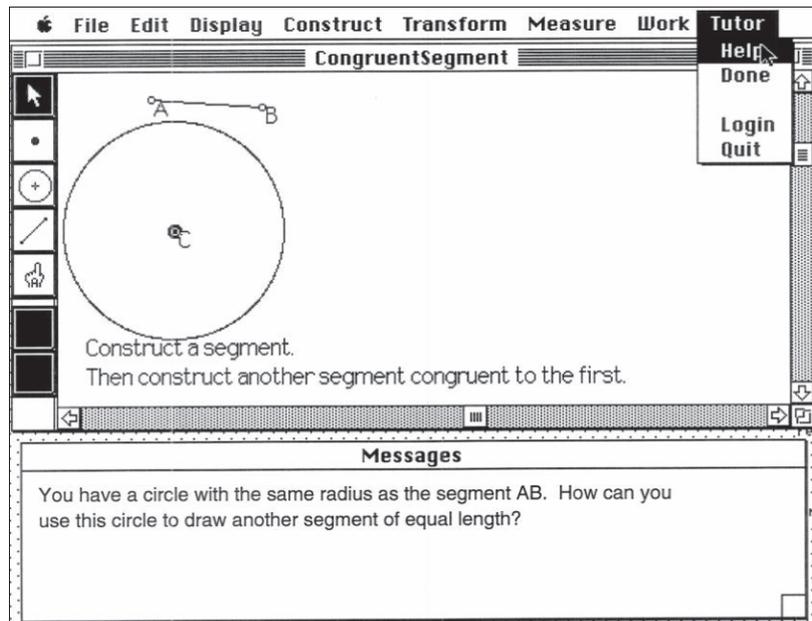


Figure 29 : Exemple d'un tuteur dans Geometer's sketchpad (Ritter and Koedinger, 1995)

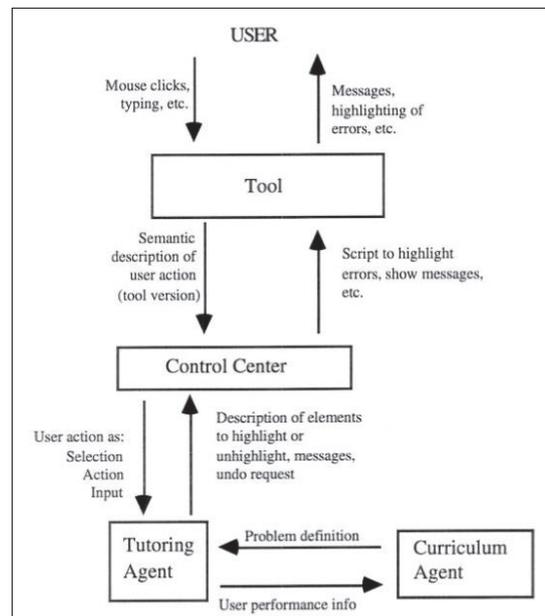


Figure 30 : Interaction entre le tuteur et le logiciel (Ritter and Koedinger, 1995)

La Figure 31 confronte les spécificités de l'approche de (Ritter and Koedinger, 1995) à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

Le modèle Lightweight Tutoring Agents adopte une approche épiphyte. Cependant, il est spécifique à l'assistance à la résolution des problèmes. De plus, la création de tuteurs avec ce modèle exige des compétences en programmation de la part des concepteurs, ainsi que l'accessibilité des composants des applications-cibles. Ainsi, dans la Figure 29, pour ajouter un item d'aide au menu de Geometer's sketchpad, il faut que l'application Geometer's sketchpad permette de le faire et que les concepteurs l'implémentent en programmant. L'absence d'un outil qui facilite la création de tels tuteurs rend cette approche inexploitable pour les enseignants non informaticiens.

Cette approche serait applicable pour créer un tuteur proposant des feedbacks pédagogiques sur les applications. Si l'assistance technique et le guidage pédagogique sont considérés comme un domaine de connaissances, les enseignants peuvent créer un tuteur qui fournit ce type d'assistance aux apprenants. Évidemment, les enseignants doivent programmer un tel tuteur.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
Lightweight Tutoring Agents	✗	✓	✓	✓	✓	✓	✓

Figure 31 : Lightweight Tutoring Agents *confronté* à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.3.2 FORMID

L'environnement FORMID (FORMation Interactive à Distance) (Guéraud *et al.*, 2009) permet la mise en œuvre de Situations Actives d'Apprentissage fondées sur l'exploitation à distance d'OPI (Objets Pédagogiques Interactifs) dans un contexte de classe virtuelle. FORMID fournit trois outils principaux. FORMID-Author permet de définir et d'ajouter des scénarios pédagogiques sur des environnements de simulation. Dans FORMID, un exercice est constitué d'un couple scénario-simulation. FORMID-Learner permet aux apprenants de réaliser un exercice en manipulant une simulation contrôlée par un scénario et de stocker les traces d'interaction de l'apprenant avec l'environnement. FORMID-Observer permet le suivi des apprenants travaillant dans FORMID.

Ici, nous détaillons FORMID-Author qui nous intéresse de par la spécification de l'assistance intégrée dans le scénario. Dans FORMID, un scénario (Guéraud *et al.*, 2004) est défini par :

- la situation initiale et l'objectif à atteindre ;
- les situations correspondant aux étapes de résolution pertinentes ; les situations particulières à observer (contraintes à respecter, erreurs classiques, dangers potentiels, etc.) ;
- la réactivité permettant d'assister l'apprenant en fonction de sa progression ; elle détermine les réactions du système (retours d'information, aide, retour au début d'étape...)

associées aux différents contrôles (étape réussie ou non, situation particulière atteinte, objectif atteint ou non).

L'exécution du scénario créé avec FORMID-Author nécessite de pouvoir contrôler l'OPI (Cortes Buitrago, 1999). Le scénario peut actuellement s'exécuter sur les simulations en Applet.

La Figure 32 confronte les spécificités du système FORMID à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

Comme Marco et Epitalk, FORMID adopte une approche épiphyte, mais il n'est pas complètement générique car il est spécifique à un type d'applications : les dispositifs de simulations.

FORMID permet de mettre en place une assistance de type scénario d'activité sur les OPI existants. Le scénario est spécifique à une activité de simulation. L'assistance est limitée à des feedbacks pédagogiques pour des activités comportant des étapes successives. FORMID n'est pas adapté pour fournir de l'assistance sur tout type d'applications-cibles, mais les enseignants peuvent spécifier des messages d'assistance technique pour les différentes étapes d'utilisation de l'application.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
FORMID	✓	✓	✓	✓	✓	✗	✓

Figure 32 : FORMID confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.4 Système de scénarisation pédagogique

Comme nous l'avons expliqué dans la section 2.1.2.3, nous avons adopté le terme de « guidage pédagogique » dans nos travaux. Cependant, le terme de « scénario pédagogique » est connu par la communauté EIAH avec une portée en général plus large. Pour cette raison, dans cette section, nous étudions les systèmes de scénarisation pédagogique.

2.3.4.1 OASIF

OASIF (Galissou and Nouveau, 2003) s'adresse aux acteurs de la FOAD (Formation Ouverte A Distance) : enseignants, ingénieurs pédagogiques, technologues. Cet outil répond à quatre objectifs principaux : représenter et décrire visuellement l'ensemble d'un dispositif de FOAD ; permettre la manipulation et la scénarisation pédagogique de ces dispositifs ; faciliter la coopération entre les principaux acteurs de la conception et de la mise en œuvre des modules de FOAD ; s'articuler avec les plates-formes logicielles de FOAD existantes.

La formation à scénariser se décompose en modules, les modules en séquences d'activités et les séquences en activités. Pour sa scénarisation, OASIF s'organise en trois principaux écrans qui permettant de définir respectivement le module, la séquence d'activités et l'activité. Il

permet donc une description structurelle. Cette description est opérationnalisable et peut donc être exécutée dans le cadre d'une FOAD concrète.

La définition de la séquence d'activités nous intéresse pour définir un guidage pédagogique dans notre contexte. OASIF permet trois types de guidage pédagogique : temporel, séquentiel et libre, qui peuvent être spécifiés au travers d'une interface graphique. Ainsi, le concepteur peut organiser l'activité dans le temps pour un guidage temporel. Il peut définir l'enchaînement purement séquentiel des activités pour un guidage séquentiel. Enfin, il peut mettre en parallèle les activités (avec un choix fait par l'apprenant) pour un guidage interactif.

Une formation définie avec OASIF peut être enregistrée au format XML permettant d'une part, la valorisation du module (présentation aux prescripteurs et futurs apprenants) et, d'autre part, son intégration dans les plateformes logicielles de FOAD (des tests ont été réalisés avec la plateforme logicielle de formation à distance de Télécom Paris (Galisson and Nouveau, 2003)).

La Figure 33 confronte les spécificités de OASIF à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

OASIF permet de définir un guidage pédagogique qui peut être intégré dans des plateformes logicielles de FOAD *via* des fichiers et non de façon épiphyte. L'utilisation des fichiers exige une interopérabilité entre le format des fichiers générés par OASIF et celui de la plateforme de FOAD. OASIF n'adopte donc pas une approche épiphyte, ni une approche complètement générique.

OASIF fournit une interface intuitive qui planifie les activités dans le temps. Cependant, il permet uniquement trois types de guidage pédagogique : temporel, séquentiel et libre.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
OASIF	✓	✗	✓	✗	✗	✓	✓

Figure 33 : OASIF confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.4.2 WebSchool

WebSchool (Balla *et al.*, 2004) est une application pour l'apprentissage en ligne. D'une part, il permet aux enseignants de produire des cours et des exercices pour les proposer aux apprenants. D'autre part, un tuteur de WebSchool peut gérer la personnalisation de l'apprentissage en utilisant les profils des apprenants.

WebSchool adopte le concept de « graphe d'unités de connaissance » pour représenter le contenu sémantique d'un cours dans un graphe où chaque sommet représente une notion du cours (ou « unité de connaissance »). Ces notions sont reliées entre elles par des arcs formalisant deux types de liens : « hiérarchiques » et « de précedence ». Les liens hiérarchiques définissent un sous-graphe de type arborescent où les sommets supérieurs représentent les notions thématiques d'un cours, alors que les feuilles de l'arborescence représentent les connaissances atomiques. Les liens de précedence établissent un ordre de parcours des notions

de même niveau par un apprenant donné. De plus, chaque unité de connaissance (ou sommet du graphe) possède une liste de notions pré-requises formalisée par une expression disjonctive de sommets.

La Figure 34 confronte les spécificités de WebSchool à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

WebSchool permet de définir un guidage pédagogique *via* un ensemble de cours grâce à un graphe d'unités de connaissances. Ce guidage est spécifique à WebSchool. Il n'est donc pas utilisable pour d'autres applications. Il n'adopte pas une approche épiphyte. Il n'exige pas de compétences de programmation.

De plus, les guidages définis restent limités à des guidages de type séquentiel et personnalisé car le graphe d'unités de connaissances n'adopte que deux types de liens : « hiérarchiques » et « de précédence ».

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
WebSchool	✓	✗	✗	✗	✗	✓	✓

Figure 34 : WebSchool confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.4.3 Telos

L'environnement Telos (Paquette, 2012) permet la définition de scénarios d'activités ou de scénarios pédagogiques multi-acteurs s'appuyant sur des ontologies. L'éditeur de scénario permet cette tâche de définition en implémentant le langage graphique MOT (Paquette, Léonard and Lundgren-Cayrol, 2007). La Figure 35 montre un exemple d'un scénario dans lequel l'activité « 5 : appliquer l'opérateur aux données » est assistée par un agent conseiller.

Au niveau du scénario pédagogique, l'enchaînement des activités est contraint par les flèches entre les activités et les conditions d'accès à ces activités. Ainsi, des activités peuvent être suivies par d'autres activités par des flèches. Le sens de la flèche reflète l'ordre des activités. D'ailleurs, Telos permet également de contraindre le passage entre activités grâce à des conditions. Une condition contient des règles de la forme <Si, Alors>. « Si » permet de consulter les activités qui précèdent cette condition ou les ressources. « Alors » permet de lancer les actions telles que « aller vers d'autres activités ». Pour permettre la définition de différents flux d'activités, il propose d'autres types de conditions regroupées en trois catégories : « séparation », « jonction », « condition événementielle / temporelle ». La condition « séparation » permet d'activer les activités qui la suivent. La condition « jonction » permet de fusionner différents flux d'activités précédentes pour activer l'activité suivante. La condition événementielle / temporelle est issue des événements (temporels) définis en entrant de cette condition. À titre d'exemple, la Figure 36 illustre la création d'une règle associée à une condition (cf. Figure 35) qui permet d'activer l'activité suivante « ajouter la trace d'exécution à la liste des solutions » si une donnée de type booléen « Le produit est une assistance du Concept-but » produite par l'activité précédente est « vrai ».

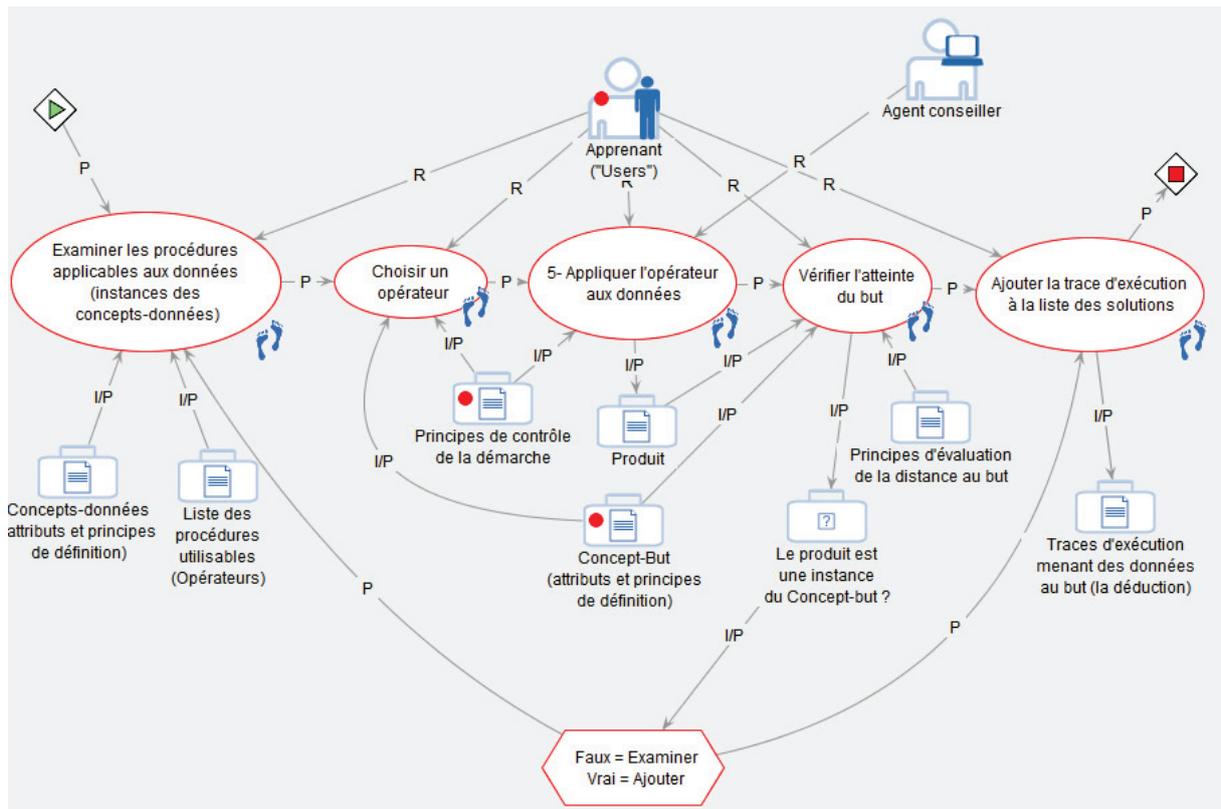


Figure 35 : Exemple d'un scénario créé par l'éditeur de scénario Telos

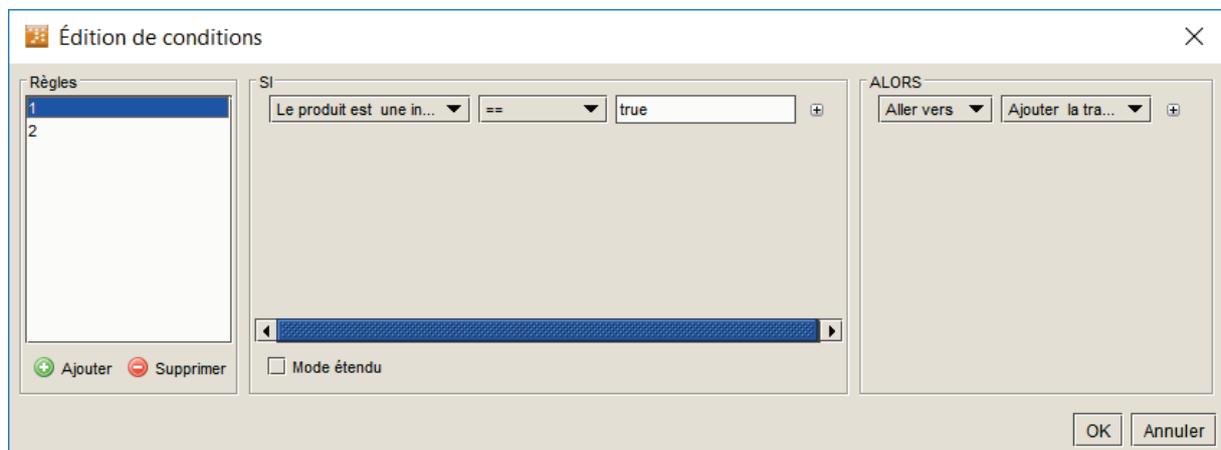


Figure 36 : Exemple de la création de règles pour une condition dans Telos

Les agents conseillers épiphytes sont associés aux activités pour assister la réalisation des activités par les acteurs. Ils contiennent des règles d'assistance de la forme <événement déclencheur, si, alors>. Les événements déclencheurs permettent de contextualiser l'assistance et sont liés à une tâche ou à une ressource du scénario, par exemple « tâche complétée » ou « ressource ouverte ». « Si » permet de connaître les conditions de déclenchement pour personnaliser l'assistance en fonction des compétences de l'acteur. « Alors » permet de lancer les actions d'assistance qui peuvent être de type modification du profil de l'acteur concerné, par exemple pour mettre à jour les compétences d'un apprenant suite à une activité ; ou de type message, affiché dans une fenêtre pop-up, par exemple pour donner un conseil à un apprenant ou notifier à un enseignant qu'un apprenant est en difficulté.

La Figure 37 montre une règle associée à un agent conseiller pour fournir un message lors de l'ouverture d'un document.

Figure 37 : Exemple de la création des règles pour un agent conseiller dans Telos

La Figure 38 confronte les spécificités de l'environnement Telos à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

Telos adopte une approche épiphyte, mais l'assistance créée est spécifique à cet environnement.

Le système Telos est très intéressant, car il permet de proposer de l'assistance pédagogique, ainsi que de l'assistance technique. Ainsi, une assistance peut afficher des messages pendant la réalisation d'une activité. Par exemple, un scénario créé avec Telos peut afficher un message qui explicite l'activité lors du lancement de l'activité ou un message qui explique l'utilisation technique d'un outil lors de son ouverture. Telos permet d'organiser les activités dans un scénario pédagogique qui peut être considéré comme un système d'assistance. En prenant en compte les ressources produites par les activités précédentes *via* des conditions, Telos permet de fournir un guidage pédagogique qui cumule les types séquentiel, contextualisé et libre.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
Telos	✓	✓	✗	✓	✓	✓	✓

Figure 38 : Telos confronté à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.4.4 Synthèse concernant les systèmes de scénarisation pédagogique

La Figure 39 confronte les spécificités globales des systèmes de scénarisation pédagogique à notre approche (épiphyte, générique, ne nécessitant pas de compétences en programmation) et aux caractéristiques de l'assistance en contexte éducatif.

Dans notre approche, nous souhaitons que la scénarisation pédagogique adopte une approche générique pour laquelle les activités d'apprentissage peuvent être réalisées sur plusieurs applications variées. Cependant, la plupart des systèmes de scénarisation pédagogique existants permettent de définir des scénarios pédagogiques au sein d'une plateforme ou d'un système spécifique donné. Ils peuvent adopter une approche épiphyte ou non. En général, ils ne demandent pas de compétences en programmation pour spécifier les contenus et l'adaptation.

Les systèmes de scénarisation pédagogique prennent en compte les différents acteurs (enseignant, tuteur, apprenant, etc.), leurs différents rôles et les différents types d'activités (activité de soutien, activité d'apprentissage). Ces activités peuvent être réalisées sur des ressources informatiques ou classiques. Le guidage pédagogique que nous souhaitons mettre en œuvre dans notre approche ne concerne que les activités d'apprentissage réalisées par les apprenants sur des ressources informatiques.

De plus, certains systèmes de scénarisation pédagogique ne proposent pas tous les types de guidages pédagogiques que nous avons identifiés.

De tels systèmes de scénarisation ne permettent donc pas une solution complète comme nous l'envisageons. Le caractère épiphyte, mais surtout l'ouverture à des applications-cibles les plus variées, permettant une approche générique est moins abordé par ces systèmes.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
Systèmes de scénarisation pédagogique	✓	✓	✓	✗	✗	✓	✓

Figure 39 : Les systèmes de scénarisation pédagogique confrontés à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.5 Hypermédias adaptatifs

De nombreux travaux portent sur les hypermédias adaptatifs. Selon (Brusilovsky, 2001), les systèmes hypermédias adaptatifs construisent un modèle des objectifs, des préférences, des connaissances de chaque utilisateur individuel, et utilisent ces modèles tout au long de l'interaction avec l'utilisateur afin de s'adapter aux besoins de cet utilisateur. (Brusilovsky, 2001) a proposé une synthèse des travaux sur les hypermédias adaptatifs. Cette synthèse met en évidence les différents contextes d'utilisation, notamment en contexte éducatif, et distingue deux zones d'adaptation : la présentation adaptative du contenu et la navigation adaptative au sein de ce contenu.

La présentation adaptative est faite par exemple dans MEDYNA (Djoudi and Behaz, 2016), un environnement numérique de travail de type hypermédia adaptatif dynamique, qui permet

d'adapter les contenus à l'apprenant en fonction de ses compétences, de son historique d'utilisation (navigation, éléments lus dans les documents) et de ses préférences. La navigation adaptative est faite par exemple dans le prototype d'hypermédia HiPPY (Miled, Laroussi and Reffay, 2015) qui aide à la navigation individualisée dynamique sur la plateforme France-IOI pour l'apprentissage de la programmation en Python.

La Figure 38 confronte les spécificités globales des systèmes hypermédias adaptatifs à notre approche (épiphyte, générique, pas de compétences de programmation) et aux caractéristiques de l'assistance en contexte éducatif.

Les systèmes hypermédias adaptatifs ne respectent pas une approche générique car ils ne proposent une adaptation à l'apprenant qu'au sein de sites web ou de dispositifs regroupant des médias interconnectés. Naturellement, ces systèmes n'adoptent pas une approche épiphyte. En général, ils ne demandent pas de compétences de programmation pour spécifier les contenus, ainsi que pour spécifier l'adaptation.

Des systèmes pour la présentation adaptative interviennent dans les contenus des ressources hypermédias (par exemple, une page web), alors que les systèmes pour la navigation adaptative interviennent dans la navigation entre des ressources. Si une activité est réalisée sur une ressource, l'assistance technique et les feedbacks pédagogiques peuvent exister sous forme d'intervention sur les contenus (par exemple, ajouter un message d'aide, alléger la page web en enlevant des contenus inutiles, etc.) et le guidage pédagogique peut exister sous forme d'intervention sur la navigation entre ressources.

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
Hypermédias adaptatifs	✓	✗	✓	✓	✓	✓	✓

Figure 40 : L'approche des hypermédias adaptatifs confrontée à notre approche et aux caractéristiques de l'assistance en contexte éducatif

2.3.6 Grafcet

Grafcet (Graphe Fonctionnel de Commande Etape/Transition) (David, 1995) est un graphe qui permet de décrire facilement les fonctionnalités d'un automatisme séquentiel. Il a été normalisé dans un premier temps par l'AFNOR (Association Française de Normalisation) en 1977 et ensuite par la CEI (Commission Electronique Internationale) en 1989.

Le fonctionnement de Grafcet (cf. Figure 41) se fait par l'enchaînement d'étapes *via* cinq règles d'évolution. Une étape peut être une étape active, une étape initiale, une macro-étape... Les actions sont associées à une étape. Le passage entre deux étapes se fait par une transition. Une transition consiste en une ou plusieurs conditions logiques (booléennes) qui traduisent la notion de réceptivité.

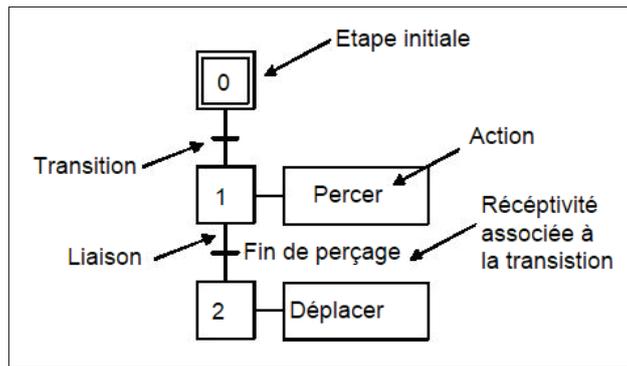


Figure 41 : Éléments principaux dans Grafcet

Les cinq règles d'évolutions de Grafcet sont les suivantes :

1. **Situation initiale** : la situation à l'instant initial est décrite par l'ensemble des étapes actives à cet instant.
2. **Franchissement d'une transition** : une transition est validée lorsque toutes les étapes, immédiatement précédentes reliées à cette transition, sont actives. Le franchissement d'une transition se produit lorsque la transition est validée et que la réceptivité associée à cette transition est vraie.
3. **Évolution des étapes actives** : le franchissement d'une transition provoque simultanément l'activation de toutes les étapes qui suivent immédiatement et la désactivation de toutes les étapes qui précèdent immédiatement.
4. **Évolutions simultanées** : plusieurs transitions simultanément franchissables sont simultanément franchies.
5. **Activation et désactivation simultanée d'une même étape** : si au cours d'une évolution, une même étape se trouve être à la fois activée et désactivée, elle reste active.

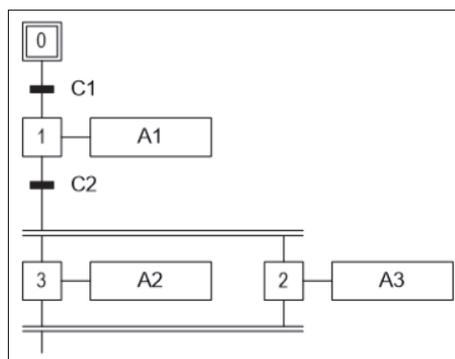


Figure 42 : Exemple de la séquence simultanée du Grafcet

Grafcet permet de créer trois types de séquences :

1. **Séquence unique** : les étapes sont réalisées les unes après les autres si elles forment une suite.
2. **Séquence simultanée** : les étapes qui suivent immédiatement une étape sont réalisées simultanément. Les étapes qui suivent le début et la fin des séquences simultanées sont représentées par un trait double et ne se trouvent pas sous les réceptivités. Ainsi, dans l'exemple de la Figure 42, les étapes 2 et 3 sont réalisées juste après l'étape 1.

3. **Séquence exclusive** : les étapes qui suivent une étape sont réalisées de manière exclusive. Pour être distinguable de la séquence simultanée, la séquence exclusive utilise un trait horizontal simple. Les étapes suivantes doivent se trouver sous les réceptivités, qui doivent donc être exclusives.

Grafcet permet d'implémenter l'idée d'un déroulement des étapes et d'explicitement trois modes de déroulement (simultané, interactif et séquentiel) grâce aux notions de transition et de réceptivité. Grafcet semble pertinent pour représenter ces déroulements, mais il n'est adapté ni au contexte de l'assistance à l'utilisateur, ni à SEPIA. Grafcet n'explicitement pas le mode indépendant et ne permet pas le mode progressif. De plus, alors que le langage aLDEAS propose des conditions avec plusieurs alternatives dans une règle, Grafcet utilise la réceptivité avec une seule alternative qui porte la valeur « vrai ».

2.4. Discussion

Dans ce chapitre, nous avons classifié les caractéristiques de l'assistance en contexte éducatif notamment (cf. Figure 43) :

- l'assistance technique pour la prise en main et l'utilisation de l'application, en exploitant la classification de l'assistance technique proposée par (Ginon, 2014) ;
- l'assistance pédagogique qui permet de proposer des feedbacks pédagogiques et un guidage pédagogique avec 5 types de guidage pédagogique : guidage libre, guidage temporel, guidage contextualisé, guidage personnalisé, guidage séquentiel ;
- le déroulement d'une assistance avec 5 modes de déroulement : indépendant, successif, simultané, progressif, interactif.

Nous avons également étudié les travaux existants proposant de l'assistance ou permettant de mettre en place une des assistances identifiées pour le contexte éducatif. Nous avons confronté ces travaux à notre classification de l'assistance ainsi qu'à notre approche pour la mise en place d'une assistance qui n'exige pas de compétences de programmation, qui adopte une approche épiphyte et générique. La Figure 44 a montré les limites de ces travaux à la vue de cette confrontation. Nous avons constaté que la plupart des travaux n'adoptent pas une approche générique. Ils sont spécifiques à une application, à un environnement, à un type d'application ou à un domaine d'application. Ainsi, le système Marco est générique par rapport aux systèmes conseillers spécifiques qui l'instancient mais il ne permet d'assister que la navigation entre des pages web. A l'inverse, certains travaux visent un objectif spécifique mais ils peuvent être exploités pour d'autres objectifs. Ainsi, le modèle Lightweight Tutoring Agents vise à construire un tuteur intelligent qui fournit principalement les feedbacks pédagogiques mais nous pouvons l'exploiter pour fournir une assistance technique. Enfin, concernant le déroulement d'une assistance, à notre connaissance, aucuns travaux n'étudient les modes de déroulement d'une assistance. Ces modes sont donc pour certains disponibles dans les systèmes mais ils ne sont donc pas explicitement représentés.

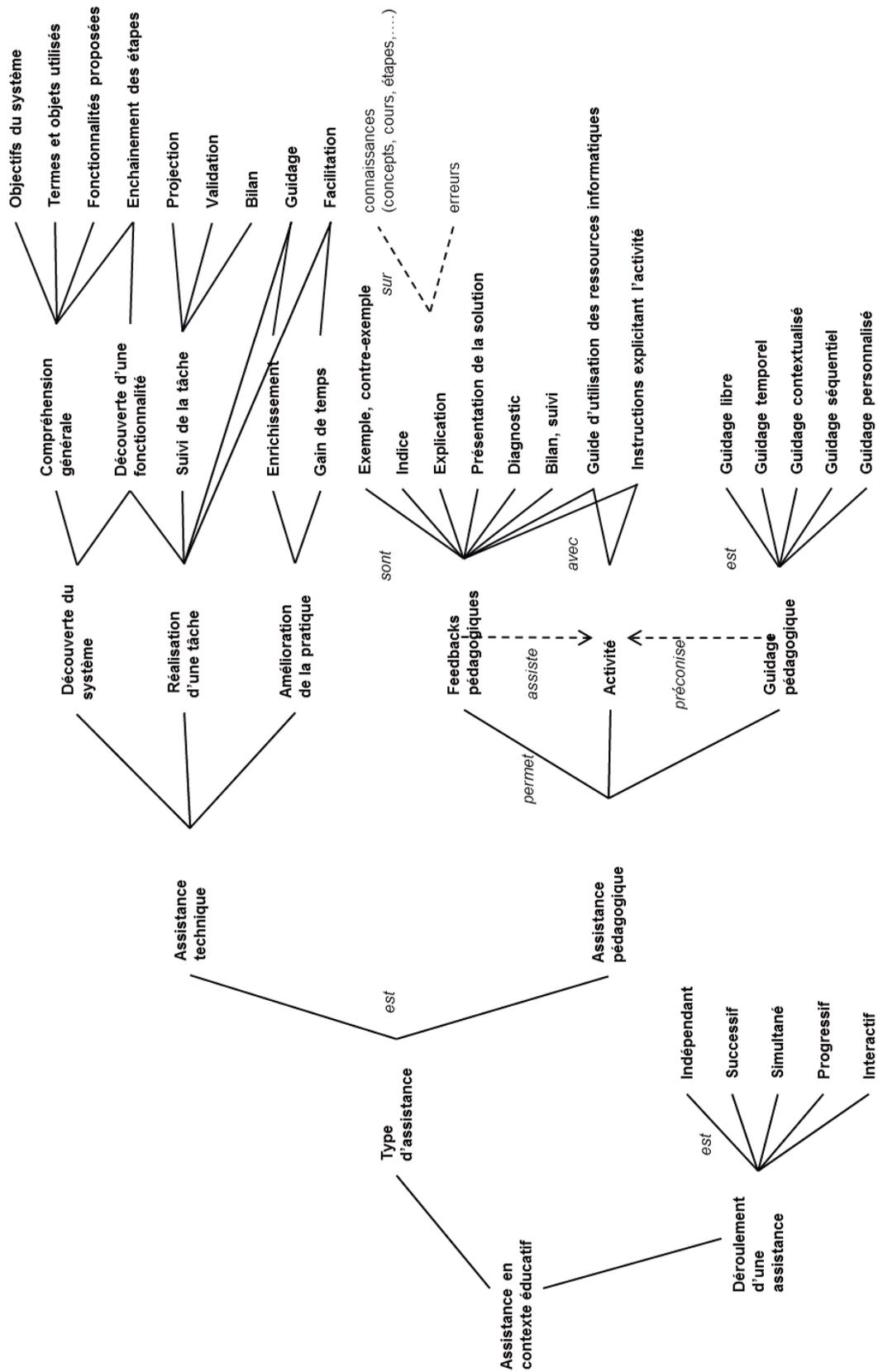


Figure 43 : Classification complète des caractéristiques d'une assistance en contexte éducatif

	Notre approche			Caractéristiques de l'assistance en contexte éducatif			
	Compétences de programmation non nécessaires	Approche épiphyte	Approche générique	Assistance technique	Feedback pédagogique	Guidage pédagogique	Déroulement d'une assistance
Marco	✓	✓	✓	✓	✓	✓	✓
Épitalk	✓	✓	✓	✓	✓	✓	✓
Hot Potatoes	✓	✗	✗	✗	✓	✗	✓
Lightweight Tutoring Agents	✗	✓	✓	✓	✓	✓	✓
FORMID	✓	✓	✓	✓	✓	✗	✓
OASIF	✓	✗	✓	✗	✗	✓	✓
Webschool	✓	✗	✗	✗	✗	✓	✓
Telos	✓	✓	✗	✓	✓	✓	✓
Hypermédiatifs adaptatifs	✓	✗	✓	✓	✓	✓	✓

Figure 44 : Les travaux connexes confrontés à notre approche et aux caractéristiques de l'assistance en contexte éducatif

En conclusion, aucun de ces travaux ne propose de solution complète pour mettre en place un système d'assistance sur des applications existantes variées en contexte éducatif.

Dans le cadre du projet AGATE, en adoptant une approche générique, les modèles et les outils ont été proposés pour mettre en place une assistance épiphyte dans les applications cibles sans modification de leur code source et sans connaissances de programmation. Ces travaux ont été conçus pour proposer une assistance technique. Nous allons dans le chapitre suivant présenté le projet AGATE, ses contributions et étudier la possibilité de les exploiter pour ajouter de l'assistance technique et pédagogique à un logiciel en contexte éducatif.

Chapitre 3. Le projet AGATE, une solution pour ajouter de l'assistance à un logiciel en contexte éducatif

Objectif de ce chapitre

Présenter les modèles et outils du projet AGATE ; confronter le langage de définition de systèmes d'assistance aLDEAS et l'éditeur d'assistances SEPIA aux caractéristiques de l'assistance en contexte éducatif.

Introduction de ce chapitre

Dans le chapitre précédent, nous avons proposé une classification des caractéristiques de l'assistance en contexte éducatif et nous avons montré que les travaux existants ne proposent pas de solution complète pour mettre en place une assistance, dans toute sa diversité, en contexte éducatif.

Ce chapitre est centré sur la présentation d'une solution pour ajouter de l'assistance à un logiciel existant en contexte éducatif. Les modèles et outils proposés au sein du projet AGATE, issus de la thèse de (Ginon, 2014), sont une base pour mettre en place une telle solution.

Précisément, dans ce chapitre, nous présentons tout d'abord le projet AGATE dans lequel s'inscrit cette thèse. Le **projet AGATE** (Approche Générique d'Assistance aux Tâches complexEs) (AGATE, 2016) propose des modèles génériques et des outils unifiés pour permettre la mise en place de systèmes d'assistance dans des applications existantes, que nous appelons applications-cibles. Les modèles et outils proposés ne sont spécifiques ni à une application ni à un domaine, mais peuvent au contraire être exploités pour ajouter une assistance à des applications les plus variées, sans que celles-ci aient été spécifiquement conçues pour permettre l'intégration d'une assistance.

Le **système SEPIA**, développé dans le cadre du projet AGATE, permet de définir de tels systèmes d'assistance en mettant en œuvre un **processus d'adjonction d'un système d'assistance sur une application-cible**. Dans SEPIA, un système d'assistance est défini par un ensemble de règles. Ces règles sont définies par des concepteurs

d'assistance grâce au **langage aLDEAS**. En fin de chapitre, **un exemple** détaillé est donné pour montrer comment définir un système d'assistance en aLDEAS avec SEPIA.

Enfin, nous **discutons** de l'exploitation de l'éditeur d'assistance SEPIA et du langage aLDEAS en contexte éducatif en les confrontant aux caractéristiques de l'assistance identifiées pour le contexte éducatif. En effet, le contexte éducatif n'est pas la cible principale du projet AGATE. Le projet aborde plus globalement l'assistance aux utilisateurs qui rencontrent des difficultés techniques dans des applications-cibles diverses.

La confrontation du système SEPIA et du langage aLDEAS aux caractéristiques de l'assistance en contexte éducatif nous amène à aborder deux questions de recherche :

Comment définir le déroulement d'une assistance avec SEPIA ?

Comment définir un guidage pédagogique avec SEPIA ?

3.5. Contexte

Le projet AGATE a pour but de proposer des modèles génériques et des outils unifiés pour rendre possible la définition de systèmes d'assistance dans des applications informatiques existantes variées. Nous qualifions ces applications d'« applications-cibles ».

Les premiers résultats de ce projet sont issus de la thèse de Blandine Ginon (Ginon, 2014) et fournissent une solution aux besoins d'assistance survenant pendant l'utilisation des applications informatiques, quelques soient ces applications et leurs utilisations.

L'approche adoptée au sein du projet AGATE consiste à mettre en place des systèmes d'assistance **épiphyte** dans les applications-cibles **sans modifier** l'application et **sans programmation**.

Un système d'assistance **épiphyte** est un système d'assistance qui se greffe sur l'application-cible sans que le fonctionnement de l'assistance ne perturbe celui de l'application-cible (Paquette, Pacht and Giroux, 1994). Le système épiphyte ne nécessite donc pas de modification de l'application-cible, ce qui rend possible l'utilisation de cette approche pour des logiciels sur lesquels nous n'avons pas la main (environnement de développement obsolète, code non disponible, etc.). Aux yeux des utilisateurs finaux, les systèmes d'assistance épiphytes paraissent toutefois intégrés à l'application-cible.

Cette approche des systèmes d'assistance épiphyte a été utilisée dans plusieurs outils, mais ces outils sont spécifiques à un environnement donné. Par exemple, Epitalk (Paquette *et al.*, 1996) est spécifique aux applications en Smalltalk-80, iFrimousse (Carlier and Renault, 2010) est spécifique aux applications Web, et les systèmes conseillers épiphytes de (Richard, 2008) sont conçus pour le Web.

Pour éviter cette limite concernant le type d'applications compatibles avec le système d'assistance, le projet AGATE adopte une approche **générique**. Les modèles et les outils proposés dans le projet ne sont donc spécifiques ni à une application, ni à un domaine d'application.

Ces modèles et outils proposés (cf. Figure 45) dans la thèse de Blandine Ginon (Ginon, 2014) permettent la mise en place de systèmes d'assistance en deux étapes : une étape de spécification du système d'assistance par le concepteur d'assistance (cf. A dans la Figure 45) et une étape d'exécution du système d'assistance à destination des utilisateurs finaux (cf. B dans la Figure 45). Le processus proposé pour l'adjonction d'un système d'assistance sur l'application-cible décrit le fonctionnement d'un système qui permet ces deux étapes. Ce processus utilise le langage aLDEAS (cf. D dans la Figure 45) comme langage pivot entre les deux étapes. Un système d'assistance est donc spécifié en langage aLDEAS et prendra la forme d'un ensemble de règles d'assistance aLDEAS. Le système SEPIA (cf. E dans la Figure 45) opérationnalise le processus d'adjonction et implémente le langage aLDEAS pour spécifier un système d'assistance.

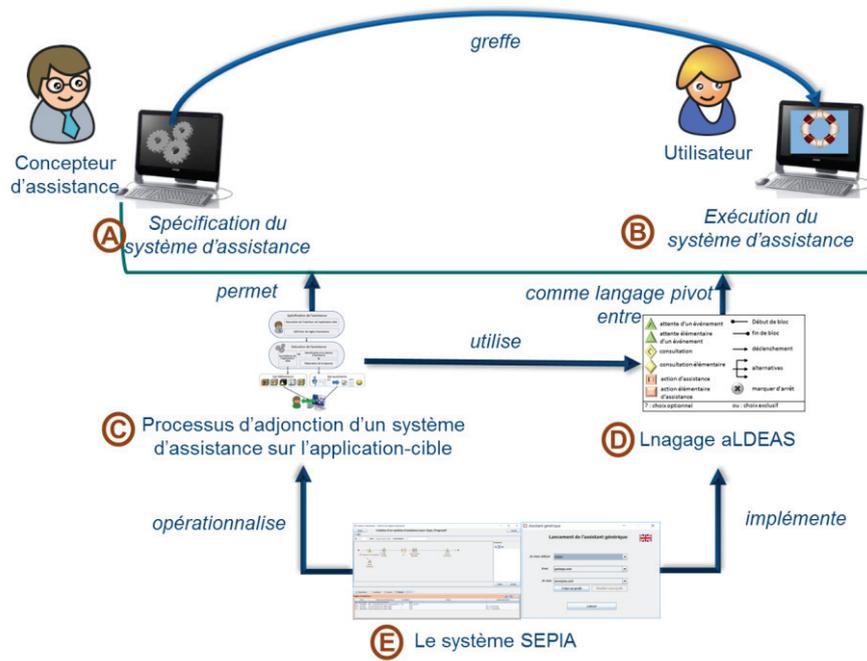


Figure 45 : Modèles et outils proposés dans le projet AGATE pour mettre en place des systèmes d'assistance épiphytes

3.5.1 Un processus d'adjonction d'un système d'assistance sur une application existante

Pour mettre en place une assistance épiphyte dans des applications-cibles, (Ginon, Jean-Daubias and Champin, 2013) ont proposé un processus d'adjonction d'un système d'assistance (cf. Figure 46). Ce processus permet aux concepteurs d'assistance qui ne sont pas nécessairement des informaticiens de définir et d'exécuter une assistance. Ce processus comporte deux phases principales : la spécification *a posteriori* d'un système d'assistance pour une application-cible donnée et l'exécution de ce système d'assistance de manière épiphyte pour les utilisateurs finaux de l'application-cible.

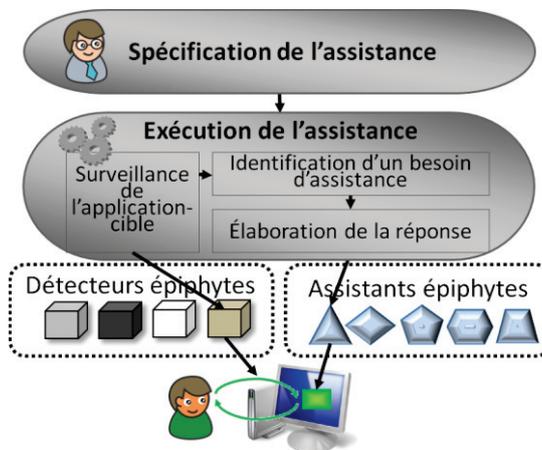


Figure 46 : Processus d'adjonction d'un système d'assistance (Ginon, Jean-Daubias and Champin, 2013)

3.5.1.1 Spécification de l'assistance

La première phase du processus d'adjonction d'un système d'assistance épiphyte est effectuée par les concepteurs d'assistance. Cette phase préparatoire permet au concepteur de spécifier l'assistance qu'il souhaite pour une application-cible. Elle comporte deux étapes : la description de l'application-cible, et la création des règles d'assistance qui définissent l'assistance souhaitée par le concepteur.

La description de l'interface de l'application-cible permet de nommer de manière unique chaque composant de l'interface. Cette description permet donc au concepteur de l'interface de connaître les composants disponibles dans l'interaction avec les utilisateurs. La description de l'interface est ensuite utilisable pour définir plusieurs assistances différentes d'une même application.

Les règles d'assistance définissent le fonctionnement du système d'assistance. La définition des règles est donc une étape clé pour pouvoir spécifier l'assistance souhaitée par le concepteur d'assistance. Pour des concepteurs non-informaticiens, cette définition par des règles d'assistance est plus accessible et moins lourde que la programmation. Autrement dit, les concepteurs d'assistance n'ont pas besoin de programmer avec cette approche.

3.5.1.2 Exécution de l'assistance

La seconde phase concerne les utilisateurs finaux de l'application-cible. Elle consiste en l'exécution de l'assistance souhaitée par le concepteur. Cette phase a lieu à chaque utilisation de l'application-cible par un utilisateur, elle est composée de trois processus : la surveillance de l'application-cible, l'identification d'un besoin d'assistance et l'élaboration de la réponse.

La surveillance de l'application permet d'observer en continu et de tracer les interactions entre l'utilisateur et l'interface de l'application-cible. Pour cela, ce processus exploite un ensemble de détecteurs épiphytes basés sur des techniques d'accessibilité. Par exemple, le détecteur basé sur Java Accessibility (Harper, Khan and Stevens, 2005) permet de surveiller les applications Java, celui basé sur UIAutomation (Haverty, 2005) se charge de surveiller les applications Windows. Le résultat de ces observations est un ensemble d'événements.

Un événement détecté est ensuite envoyé au processus d'identification d'un besoin d'assistance qui s'exécute en parallèle. Ce processus parcourt toutes les règles définies. Il lance les règles dont l'événement déclencheur correspond à l'événement détecté. L'exécution de ces règles permet d'identifier un besoin d'assistance. Lorsqu'un besoin d'assistance est identifié, le processus d'élaboration d'une réponse à ce besoin identifié est déclenché.

Le processus d'élaboration de la réponse fournit une réponse sous forme d'une ou plusieurs actions d'assistance, réalisées dans l'application-cible par des assistants épiphytes. Ces assistants épiphytes sont variés afin de pouvoir assister au mieux les différents utilisateurs. Par exemple, l'action d'assistance de type mise en valeur (entourer, colorer un bouton, etc.) permet d'attirer l'attention de l'utilisateur sur un élément de l'écran.

Ce processus d'adjonction d'un système d'assistance à une application-cible existante adopte une approche à la fois épiphyte et générique. Il est structuré en deux phases : la spécification et l'exécution de l'assistance. Pour faire le lien entre ces deux phases, le langage pivot aLDEAS a été proposé dans le projet AGATE (Ginon, Jean-Daubias, *et al.*, 2014a). Ce

langage, qui est présenté dans le chapitre suivante, permet de définir dans la phase de spécification un système d'assistance sous forme d'un ensemble de règles aLDEAS qui sont ensuite exploitées sur l'application-cible dans la phase d'exécution.

3.5.2 Le Langage aLDEAS

aLDEAS (a Language to Define Epi-Assistance Systems) est un langage graphique qui permet la définition de systèmes d'assistance pour des applications existantes sous forme d'un ensemble de règles (Ginon, Jean-Daubias, *et al.*, 2014a). Ce formalisme est au cœur du processus d'adjonction de systèmes d'assistance épiphytes : au travers de sa mise en œuvre, il permet au concepteur de spécifier l'assistance souhaitée lors de la phase de spécification de l'assistance, puis il est exploité lors de la phase d'exécution de l'assistance. aLDEAS est constitué de différents types de composants qui peuvent être combinés pour créer des règles et actions d'assistance, afin de répondre à des besoins d'assistance variés.

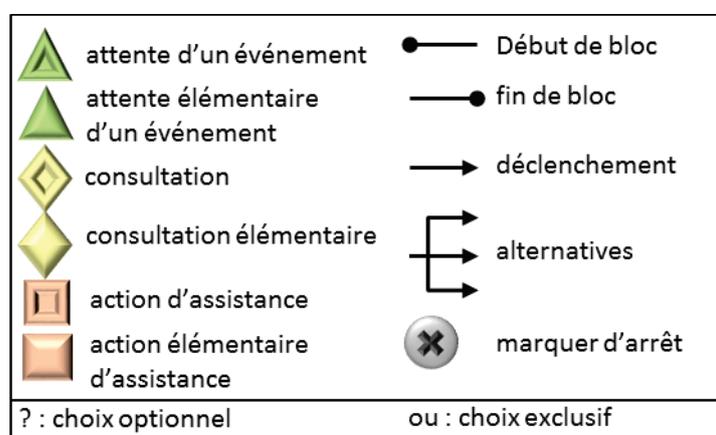


Figure 47 : Éléments de base aLDEAS

Les principaux éléments d'aLDEAS sont les attentes d'événements sur l'application-cible, les consultations (de l'utilisateur, de l'historique de l'assistance, du profil de l'utilisateur ou des traces de l'utilisateur) et les actions d'assistance (message, mise en valeur, etc.).

3.5.2.1 Attente d'événement

Les **attentes d'événements**  rendent le système d'assistance réactif et à l'écoute des événements qui surviennent, d'une part dans l'application-cible, et d'autre part au niveau de l'assistance. Ainsi, les attentes d'événement permettent d'attendre qu'un événement donné ait lieu sur une application-cible pour déclencher un autre élément qui suit une flèche de « déclenchement »  .

Les événements attendus peuvent être des **événements élémentaires**  qui concernent des actions de l'utilisateur (clic sur un composant, sélection de texte, etc.), de l'assistance (lancement de l'assistance, déclenchement d'une règle, etc.), ou la durée (une durée après un événement, une durée après l'absence d'un événement, etc.). La totalité des types d'événements élémentaires proposés dans le langage ALDEAS sont présentés dans une typologie détaillée en Annexe C (cf. page 212).

Les événements attendus peuvent également être des **événements composés** , c'est-à-dire une combinaison de plusieurs événements élémentaires. Les attentes d'événements composés sont utilisées pour représenter la réalisation d'une tâche de « haut-niveau » telle que l'attente d'événement E_1 « sauvegarder un document Word ». Cet événement composé est constitué de plusieurs actions de « bas-niveau » : clic sur « enregistrement », choix du dossier qui le contiendra, nommage de ce document, clic sur le bouton OK (cf. Figure 48).

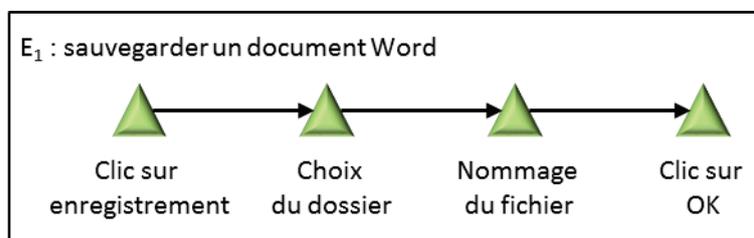


Figure 48 : Exemple d'une attente d'un événement composés E_1 « sauvegarder un document Word »

Pour simplifier le discours dans la suite du manuscrit, pour parler d'« attente d'événement » tel que décrit dans le langage aLDEAS, nous utiliserons simplement le terme « événement ».

3.5.2.2 Consultation

Les **consultations**  permettent de personnaliser et de contextualiser l'assistance en conditionnant les actions du système d'assistance à des informations relatives aux actions passées de l'utilisateur, à son profil, à ses choix et à l'état de l'application-cible. Une consultation est suivie de flèches proposant plusieurs alternatives , chacune associée à un test dont le type dépend du type de la valeur renvoyée par cette consultation. Par exemple pour un texte, les tests peuvent imposer qu'il soit égal, qu'il contienne, commence ou se termine par un texte donné.

Les consultations peuvent être des **consultations élémentaires**  : consultation de l'utilisateur (c'est-à-dire un choix de l'utilisateur pour personnaliser l'assistance), consultation de l'état de l'application-cible (valeur d'un composant, est-il actif, etc.) ou consultation liée à l'assistance (du profil de l'utilisateur, de l'historique de l'assistance, etc.). La totalité des types de consultations élémentaires proposés dans le langage ALDEAS sont présentés dans une typologie détaillée en Annexe C (cf. page 212).

Les consultations peuvent être des **consultations composées**  lorsqu'elles combinent plusieurs consultations élémentaires grâce à une formule logique.

3.5.2.3 Action d'assistance

Afin de répondre efficacement et de manière personnalisée aux besoins d'assistance les plus variés des utilisateurs, un système d'assistance doit être capable de fournir des actions d'assistance riches et diversifiées. Le langage aLDEAS propose un large choix d'**actions élémentaires d'assistance**  : messages, mises en valeur ou actions automatisées sur un composant de l'interface de l'application-cible, propositions de ressources externes (comme

des vidéos de démonstration, forum, supports de cours), etc. La totalité des types d’actions d’assistance élémentaires proposés dans la langage aLDEAS sont présentés en Annexe C (cf. page 212).

Les éléments aLDEAS peuvent être combinés pour former un bloc d’éléments correspondant à des **actions composées d’assistance** . Une action composée est un bloc aLDEAS délimité par les symboles  et , le symbole  déclenchant l’élément suivant. Un marqueur d’arrêt  qui suit une attente d’événement met fin à toutes les actions d’assistance déclenchées depuis le début du bloc. Par exemple, une action composée (cf. Figure 49) débute par une action élémentaire « afficher un message ». Ce message est fermé après 10 secondes grâce au marqueur d’arrêt.

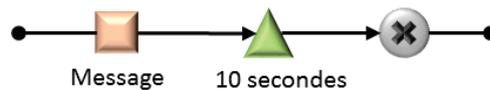


Figure 49 : Exemple d’une action composée

Pour faciliter la définition d’une action composée, aLDEAS propose les patrons d’actions composées notamment le patron de règles aLDEAS.

3.5.2.4 Patron de règles aLDEAS

Le **patron de règles** (cf. Figure 50) permet aux concepteurs de spécifier, par un ensemble de règles, l’assistance qu’ils souhaitent pour une application-cible.

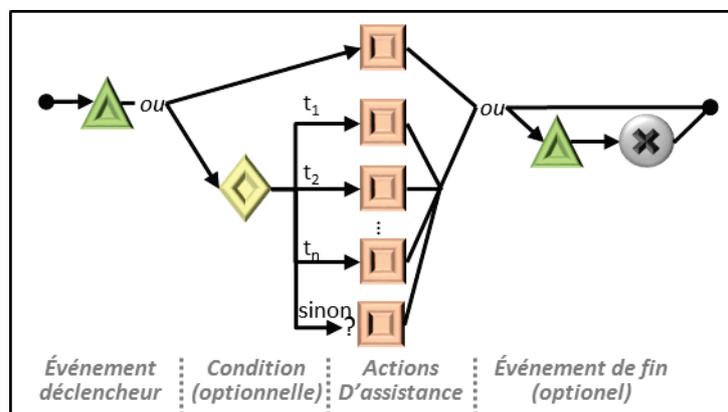


Figure 50 : Patron de règles aLDEAS

Le patron de règles, au cœur du langage aLDEAS, permet de formaliser une action composée d’assistance. Le fonctionnement du système d’assistance est représenté par l’enchaînement des règles aLDEAS qui respecte le patron de règles. Dans ce patron, tous les éléments sont des éléments composés, car un élément composé peut également représenter un élément élémentaire.

Une règle aLDEAS débute par l’attente d’un événement nommé événement déclencheur. Dès que cet événement a lieu, le lancement de la ou des actions d’assistance est soit immédiat (chemin du haut sur la Figure 50) soit contraint par une condition (chemin du bas sur la Figure 50). Cette condition prend la forme d’une consultation avec une alternative associée à chacune

de ces actions d'assistance. Les actions lancées peuvent être des actions élémentaires d'assistance ou des règles. Enfin, la règle peut se terminer par un événement de fin suivi par le marqueur d'arrêt qui met fin à toutes les actions élémentaires déclenchées par cette règle. Dans le cas où la règle ne contient aucun événement de fin, cette règle ne prend pas fin et les actions de cette règle ne sont pas clôturées par un événement de fin.

Prenons l'exemple de la règle présentée sur la Figure 51, cette règle fait partie d'un ensemble de règles qui constitue un système d'assistance pour le site Web ActiveMath (Melis *et al.*, 2001). Elle vérifie la réponse de l'apprenant. Ainsi, la règle débute par un événement déclencheur qui consiste en une action de l'utilisateur : un clic sur le bouton d'aide. Dès que cet événement a lieu (l'apprenant a besoin d'aide et demande explicitement de l'assistance), cette règle est lancée et vérifie la réponse en consultant le texte écrit par l'apprenant. Il s'agit d'une consultation de l'état de l'application-cible. Si le texte n'est pas égal à 1 (c'est-à-dire que l'apprenant n'a pas saisi la bonne réponse), cette règle affiche un message pour aider l'apprenant à se corriger. Ce message est affiché pendant seulement 10 secondes grâce à l'événement de fin qui est une durée.

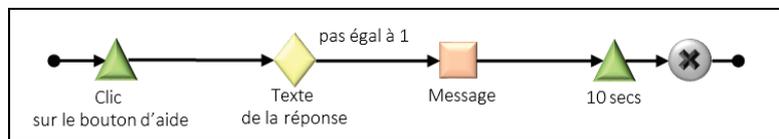


Figure 51 : Exemple d'une règle sur ActiveMath

Avec le patron de règles, un concepteur peut définir un système d'assistance. En outre, le langage aLDEAS a été également complété par d'autres patrons qui facilitent la définition d'actions d'assistance (Ginon, Jean-Daubias, *et al.*, 2014a) : un patron permettant de mettre en place une assistance pas à pas et un permettant de faire une présentation guidée de l'application ou des fonctionnalités.

Le langage aLDEAS, les patrons et le processus d'adjonction d'un système ont été implémentés dans le système SEPIA qui est présenté dans le chapitre suivant.

3.5.3 Le système SEPIA

Afin de mettre en œuvre les différentes contributions théoriques du projet AGATE présentées dans les chapitres précédente, (Ginon, Jean-Daubias, *et al.*, 2014b) ont développé le système SEPIA (Specification and Execution of Personalized Intelligent Assistance) dont l'architecture est présentée sur la Figure 52.

Nous pouvons y voir que SEPIA se compose de deux outils principaux : l'éditeur d'assistance qui permet aux concepteurs d'assistance de spécifier une assistance sous forme d'un ensemble de règles aLDEAS et le moteur d'assistance qui exécute ces règles pour fournir l'assistance.

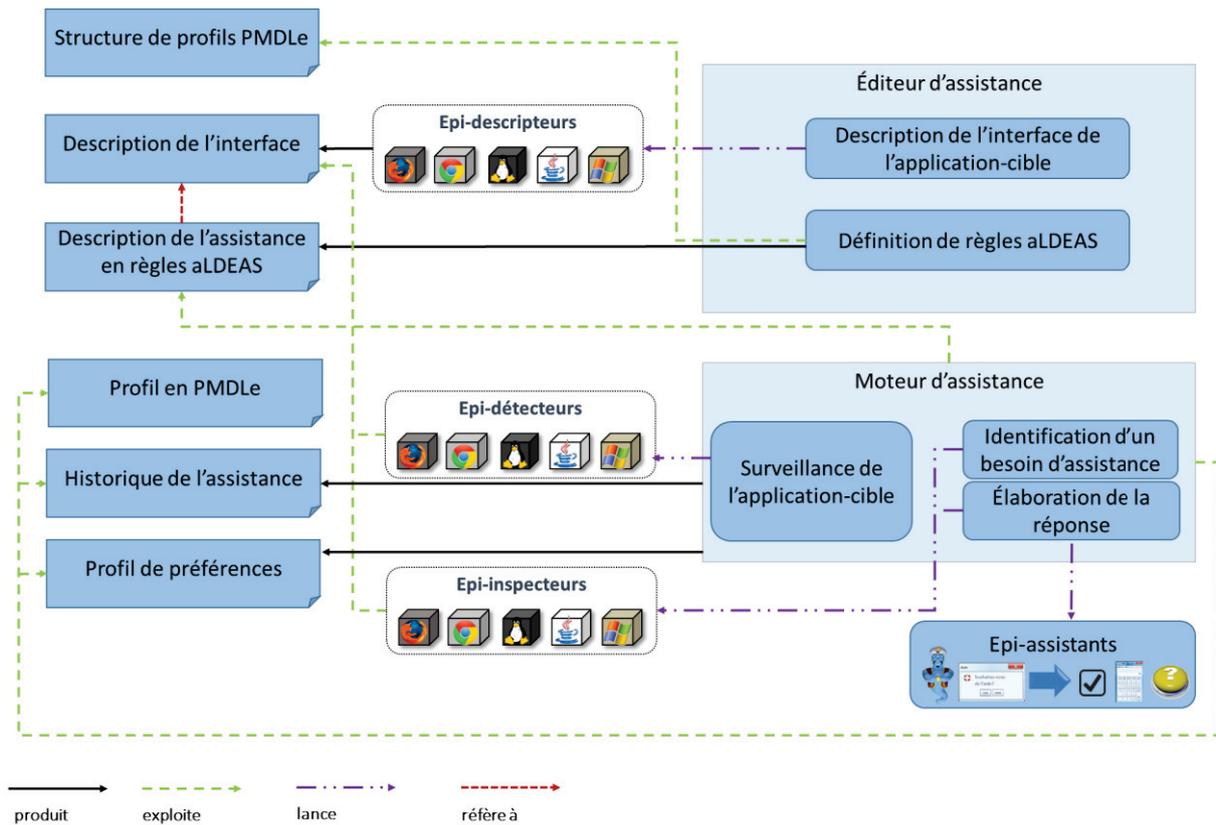


Figure 52 : Architecture du système SEPIA (Ginon, 2014)

3.5.3.1 Éditeur d'assistance

L'éditeur d'assistance (cf. Figure 53) s'adresse aux concepteurs d'assistance et met en œuvre la première phase du processus d'adjonction d'un système d'assistance présenté dans la section 3.5.1.

Il fournit un ensemble des descripteurs épiphytes pour décrire l'interface de plusieurs types d'applications (Windows, Java, MacOS et Web). L'identification des composants se fait grâce à leur hiérarchie dans l'interface (Ginon, Champin and Jean-Daubias, 2013). La description de l'interface est formalisée en XML et stockée dans un fichier qui peut être réutilisé pour différentes assistances.

La définition de règles est une des fonctionnalités principales de l'éditeur d'assistance. Pour cette étape (cf. Figure 53), deux interfaces, l'une textuelle, l'autre graphique, sont proposées. Le concepteur d'assistance peut définir une règle en glissant-déposant des éléments aLDEAS à l'écran. Lors de cette étape, le concepteur d'assistance peut créer des éléments aLDEAS concernant les composants de l'interface de l'application-cible (un événement de l'action de l'utilisateur, la consultation de l'état de l'application, la mise en valeur d'un composant, etc.). Pour exploiter le profil de l'utilisateur, SEPIA adopte le langage PMDLe (Eyssautier-Bavay, 2008; Jean-Daubias, Ginon and Lefevre, 2011). Il est donc nécessaire dans ce cas de disposer de la structure des profils avant de pouvoir définir les règles.

L'ensemble des règles définies compose la description de l'assistance en aLDEAS. Cette description est formalisée en XML et stockée dans un fichier XML qui peut être ensuite exécuté par le moteur d'assistance.

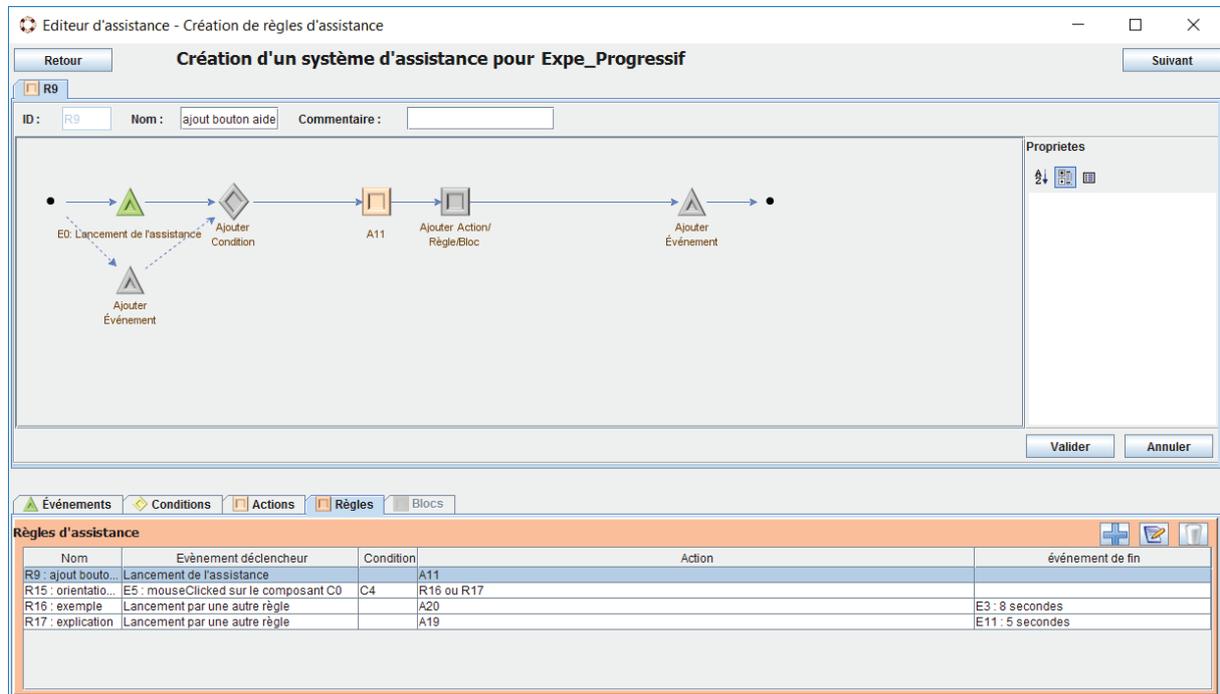


Figure 53 : Écran principal de l'éditeur d'assistance de SEPIA

3.5.3.2 Moteur d'assistance

Le moteur d'assistance concerne, lui, les utilisateurs des applications-cibles. Il permet d'exécuter les systèmes d'assistance préalablement conçus par le concepteur de l'assistance. Le moteur exploite en effet la description de l'assistance en règles aLDEAS définies avec l'éditeur d'assistance.

Actuellement, les détecteurs épiphytes développés permettent de surveiller les types d'applications suivants : Windows, Java, Web, Linux, MacOS, et les inspecteurs épiphytes développés permettent de consulter l'état des applications de ces mêmes types. Un événement détecté, l'état de l'application consultée, le profil de l'apprenant en PMDLa La (niveau d'étude, maîtrise d'une connaissance, etc.), l'historique de l'assistance qui contient toutes les informations relatives à l'exécution d'un système d'assistance (nombre de lancements d'une règle donnée) permettent l'identification d'un besoin d'assistance.

Dans SEPIA, les assistants épiphytes développés permettent de fournir de l'assistance aux utilisateurs sous des formes variées : message, mise en valeur ou masquage d'un élément d'interface, ajout d'un composant, proposition de ressources externes, animation. Pour fournir aux assistants épiphytes les coordonnées des composants sur l'interface de l'application-cible, le processus d'élaboration de la réponse exploite également les inspecteurs épiphytes. Les assistants épiphytes exploitent également les inspecteurs. Ainsi, dans l'exemple de la Figure 54, le moteur ajoute des symboles et un bouton à côté des composants de l'interface de l'application-cible. L'assistant connaît la position des composants grâce à l'inspecteur Web. Tous les fichiers exploités sont formalisés en XML. Le moteur d'assistance permet aux

utilisateurs finaux de manipuler le profil de préférences en PMDLe (couleur ou police préférée pour le texte par exemple) qui permettra de compléter l'élaboration de la réponse.

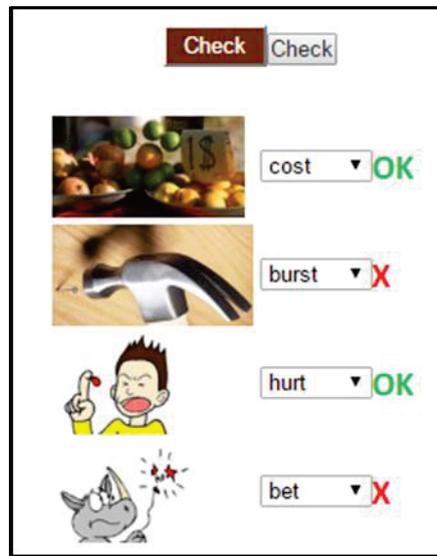


Figure 54 : Ajout d'un bouton et de symboles sur l'interface de EOLF à l'aide de SEPIA

Actuellement, il y a deux versions du moteur d'assistance : la version bureau et la version Web. La version bureau permet d'exécuter les systèmes d'assistance sur les applications Windows, Java, Web et MacOS, alors que la version Web permet d'exécuter les systèmes d'assistance uniquement sur les applications Web. La version Web du moteur SEPIA a certes une couverture plus faible des types applications, mais elle est utilisable directement, sans installation du moteur. Les utilisateurs doivent seulement installer une extension dans le navigateur Chrome qui est chargée du détecteur Web et des assistants épiphytes. De plus, l'exécution des systèmes d'assistance avec la version Web du moteur est plus rapide.

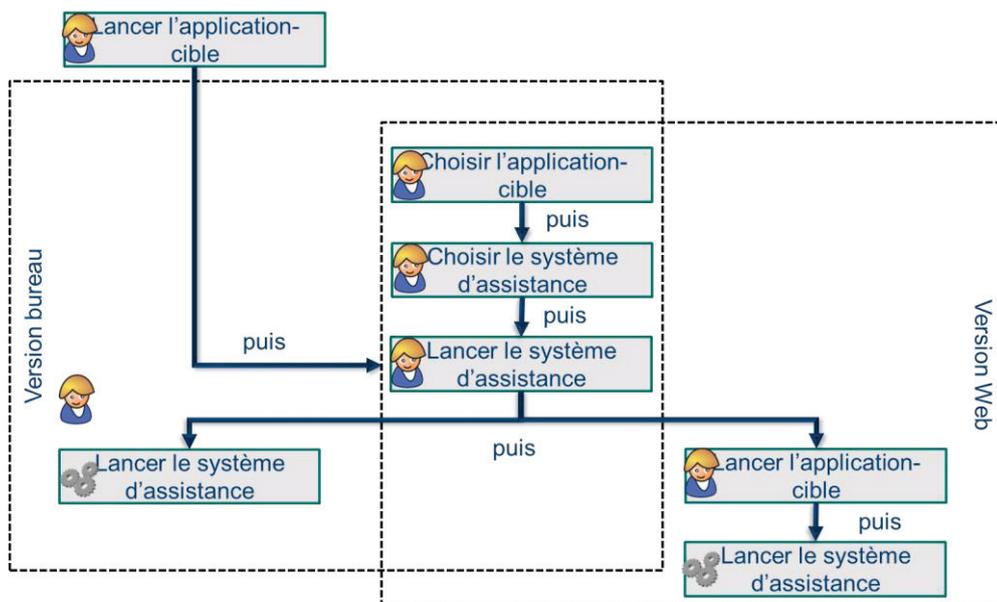


Figure 55 : Étapes du lancement d'un système d'assistance selon les deux versions du moteur d'assistance SEPIA

La Figure 55 illustre les étapes du lancement d'un système d'assistance selon les deux versions du moteur d'assistance. Pour lancer un système d'assistance avec le moteur SEPIA, les utilisateurs doivent suivre les étapes suivantes : choisir l'application-cible à assister, choisir le système d'assistance souhaité, puis lancer ce système d'assistance.

L'application-cible doit être ouverte avant le lancement du système d'assistance par le moteur. Ainsi, lors du lancement de l'assistance par le moteur, certaines règles du système d'assistance sont lancées. L'utilisateur peut donc manquer ces règles pour recevoir de l'assistance.

Avec la version bureau du moteur, lors du lancement du système d'assistance par l'utilisateur, ce système est immédiatement lancé par le moteur. L'utilisateur doit donc lancer l'application-cible avant le lancement du système d'assistance par l'utilisateur.

Comment une version améliorée, la version Web du moteur force les utilisateurs à lancer l'application-cible avant le lancement de l'assistance par le moteur. Ainsi, lors du lancement de l'assistance par l'utilisateur, le moteur demande aux utilisateurs de lancer l'application-cible. Après avoir détecté le lancement de l'application-cible, l'assistance est lancée effectivement par le moteur.

3.5.4 Exemple d'un système d'assistance en aLDEAS

Dans cette section, nous donnons l'exemple d'un système d'assistance défini avec l'éditeur SEPIA. Ce système est donc constitué d'un ensemble de règles respectant le patron de règles du langage aLDEAS.

Notre exemple concerne PhotoScape, un logiciel de retouche d'images. Il fournit des fonctionnalités très variées, notamment une permettant de corriger les yeux rouges sur une photo. Cependant, les nouveaux utilisateurs n'arrivent pas toujours à utiliser cette fonctionnalité. Dans le cadre du projet AGATE, nous avons donc ajouté un système d'assistance à PhotoScape par l'intermédiaire de SEPIA pour aider ces utilisateurs (Ginon, 2014). Ce système d'assistance guide l'utilisateur au sein du logiciel pour lui montrer comment utiliser la fonctionnalité de correction des yeux rouges. L'exécution de ce système d'assistance est illustrée par une vidéo disponible sur le site Web du projet AGATE¹ (cf. Figure 56).

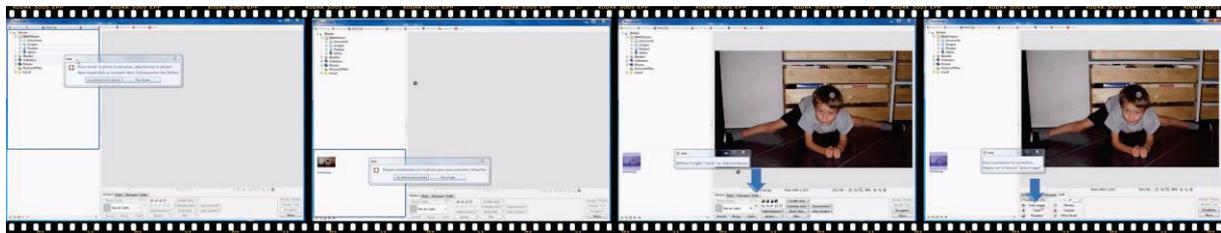


Figure 56 : Vue d'ensemble de l'exemple de système d'assistance qui guide l'utilisateur pour la correction de yeux rouges

¹ Lien de la vidéo de l'exécution du système d'assistance de Photoscape : https://www.youtube.com/watch?v=OOeSg_upxk

Nous avons choisi cet exemple notamment parce qu'il montre que SEPIA propose des solutions pour définir le système d'assistance malgré certaines limites techniques. Par exemple, certains événements de certains composants de PhotoScape ne sont pas détectables par SEPIA.

Nous allons à présent détailler le contenu de ce système d'assistance, son déroulement et montrer pour chaque étape comment il est mis en œuvre à l'aide des règles définies selon le langage aLDEAS.

- Étape 1 (cf. Figure 57) : après le lancement de PhotoScape et lors du lancement de l'assistance, le système d'assistance demande à l'utilisateur s'il souhaite de l'aide. Si l'utilisateur répond oui, le système d'assistance effectue l'étape 2, sinon, il s'arrête.

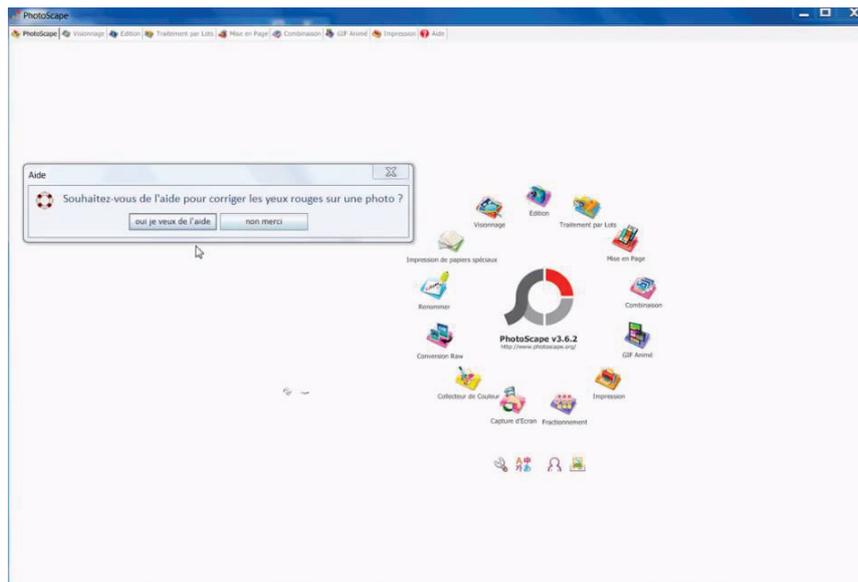


Figure 57 : Exécution de l'étape "demande si l'utilisateur souhaite de l'aide" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS :

R_0 (cf. Figure 58) : La règle R_0 débute par un événement déclencheur « lancement de l'assistance ». Elle demande à l'utilisateur s'il souhaite de l'aide *via* la consultation utilisateur C_0 . Si l'utilisateur clique sur le bouton « oui je veux de l'aide », R_0 lance la règle R_1 pour effectuer l'étape 2.

C_0 : type de consultation = « consultation utilisateur », texte = « Souhaitez-vous de l'aide pour corriger les yeux rouges sur une photo ? », choix = « oui je veux de l'aide / non merci »

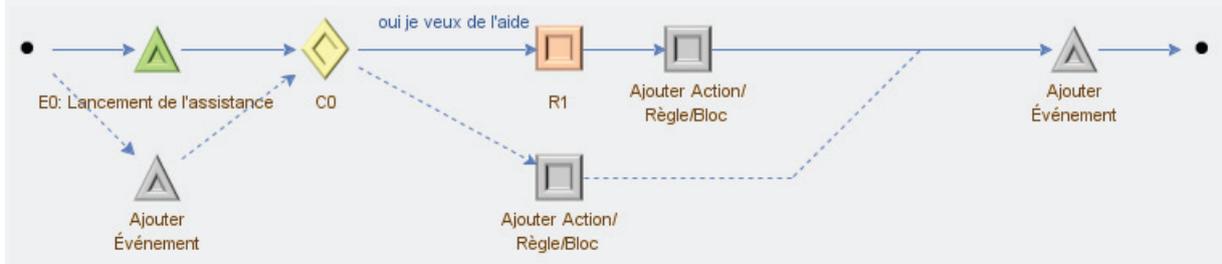


Figure 58 : Définition de la règle R_0 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

- Étape 2 (cf. Figure 59) : lorsque l'utilisateur a répondu par l'affirmative dans l'étape 1, le système d'assistance lui demande de cliquer sur l'onglet « Édition » où se trouve la fonctionnalité de correction des yeux rouges.

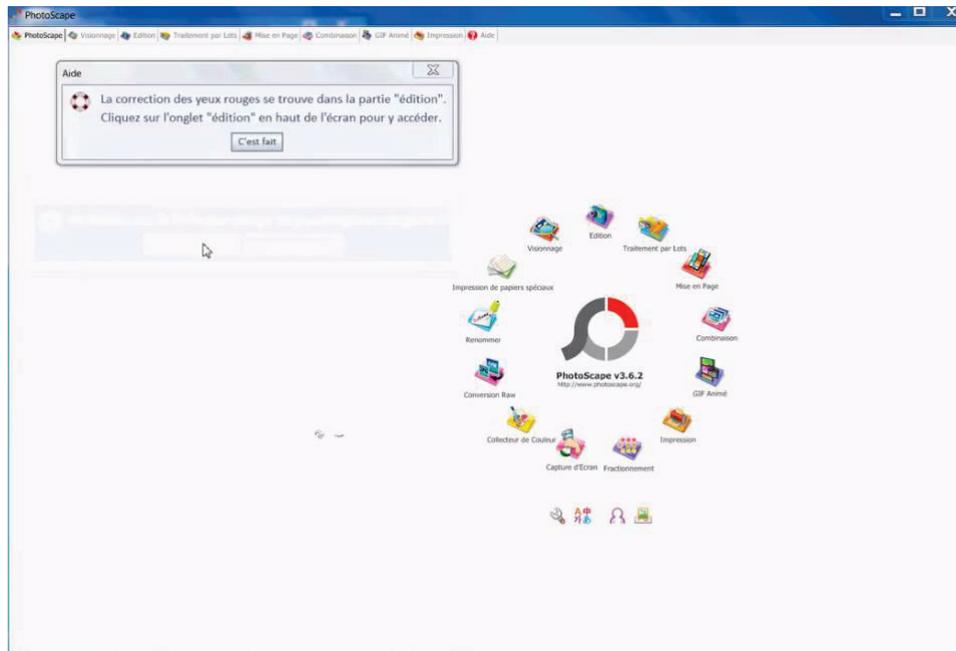


Figure 59 : Exécution de l'étape "demande de clic sur édition" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS :

R_1 (cf. Figure 60) : La règle R_1 débute par l'événement déclencheur « lancement par une règle ». Dans cette étape, la règle qui lance R_1 est la règle R_0 présentée dans l'étape 1. Comme le détecteur Windows de SEPIA ne peut pas détecter le clic par l'utilisateur sur l'onglet « édition » dans PhotoScape, R_1 demande à l'utilisateur s'il a déjà cliqué sur cet onglet *via* la consultation utilisateur C_1 . Le cas échéant, R_1 lance R_2 pour effectuer l'étape 3.

C_1 : type de consultation = « consultation utilisateur », texte = « La correction des yeux rouges se trouve dans la partie "édition". Cliquez sur l'onglet "édition" en haut de l'écran pour y accéder. », choix = « C'est fait »

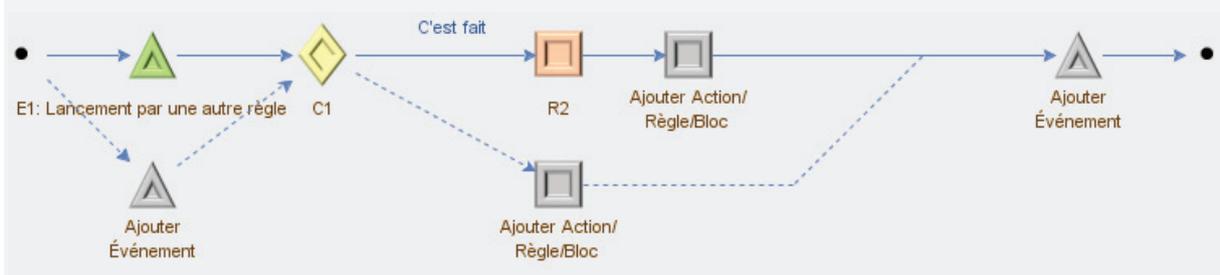


Figure 60 : Définition de la règle R_1 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

- Étape 3 (cf. Figure 61) : une fois que l'utilisateur a indiqué que l'action précédente est effectuée, le système entoure l'arborescence des dossiers de l'onglet « édition » pendant 5 secondes, lui demande d'y sélectionner le dossier qui contient la photo à corriger et lui

demande s'il veut plus d'aide. S'il veut plus d'aide, le système effectue l'étape 4a, sinon l'étape 4b.

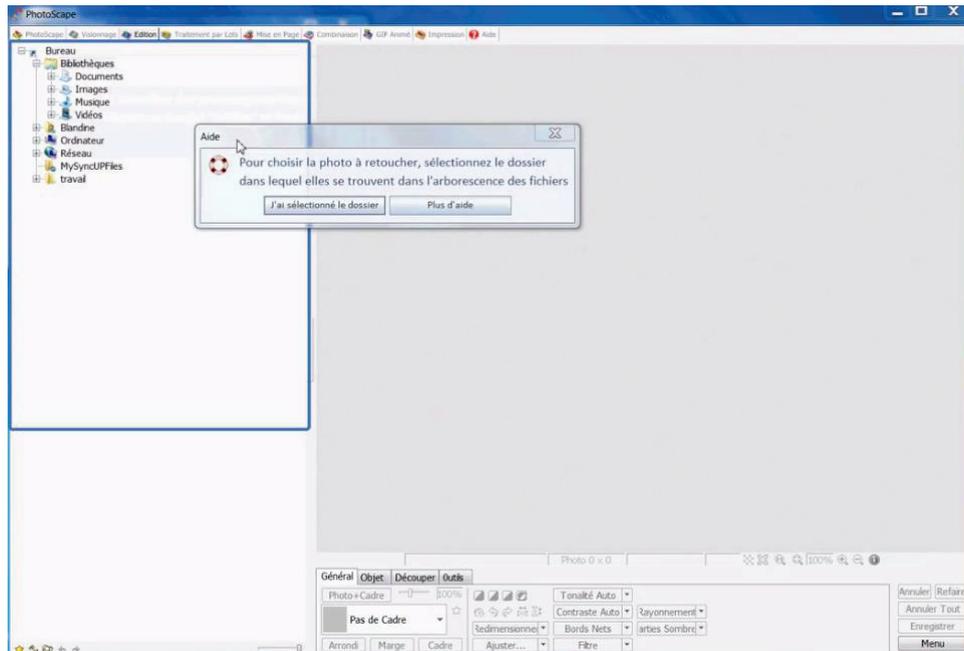


Figure 61 : Exécution de l'étape "demande de la sélection d'un dossier" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS :

R_2 (cf. Figure 62) : La règle R_2 permet d'entourer l'arborescence des dossiers. Comme nous l'avons expliqué dans l'étape précédente, la détection du clic sur l'onglet « édition » est remplacée par une consultation de l'utilisateur C_1 dans la règle R_1 . R_2 débute donc par l'événement déclencheur « lancement par une règle » au lieu d'un événement clic sur « édition ». R_2 effectue l'action d'assistance A_0 , puis lance la règle R_3 .

A_0 : type d'action = « mise en valeur », type de mise en valeur = « entourer », composant = « 130 » (arborescence des dossiers)

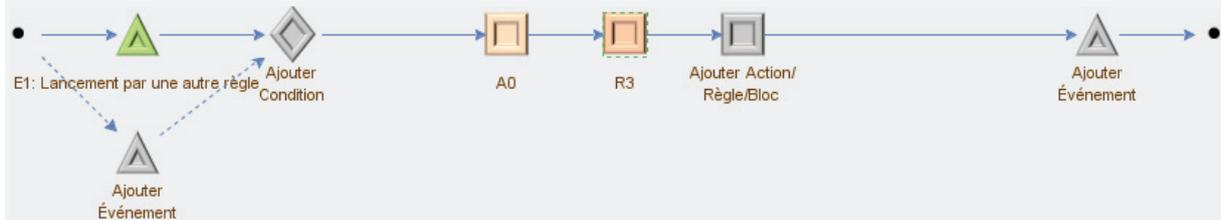


Figure 62 : Définition de la règle R_2 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

R_3 (cf. Figure 63) : La règle R_3 permet d'afficher un message à l'utilisateur pour lui demander de sélectionner le dossier et lui demander s'il veut plus d'aide. R_3 débute par l'événement déclencheur « lancement par une règle ». Les demandes faites à l'utilisateur dans la règle R_3 sont effectuées *via* la consultation utilisateur C_2 . En fonction de la réponse de l'utilisateur,

l'action d'assistance A_1 et la règle R_5 (respectivement A_2 et R_4) sont lancées pour effectuer l'étape suivante 4a (respectivement l'étape 4b).

C_2 : type de consultation = « consultation utilisateur », texte = « Pour choisir la photo à retoucher, sélectionnez le dossier dans lequel elle se trouve dans l'arborescence des fichiers », choix = « j'ai sélectionné le dossier / plus d'aide »

A_1 : type d'action « mise en valeur », type de mise en valeur = « entourer », composant = « 220 » (liste de photos)

A_2 : type d'action = « mise en valeur », type de mise en valeur = « afficher un symbole », symbole = « flèche », direction = « par la droite », composant = « 137 » (arbre de dossiers)

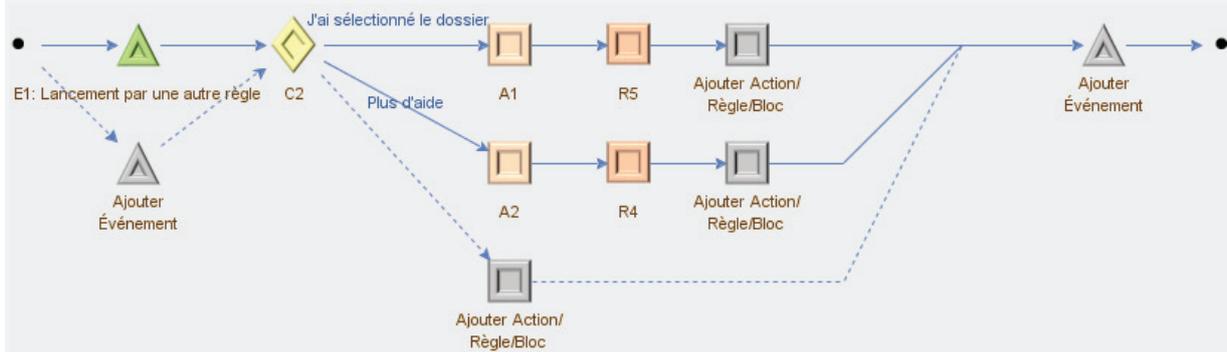


Figure 63 : Définition de la règle R_3 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

- Étape 4a (cf. Figure 64) : lorsque l'utilisateur demande plus d'aide à l'étape 3, le système essaie une autre forme de mise en valeur de l'arborescence de dossiers en ajoutant une flèche pointant vers le composant concerné pour attirer l'attention de l'utilisateur. Le système demande de nouveau à l'utilisateur de sélectionner le dossier dans cette arborescence, mais de façon plus précise et explicite.

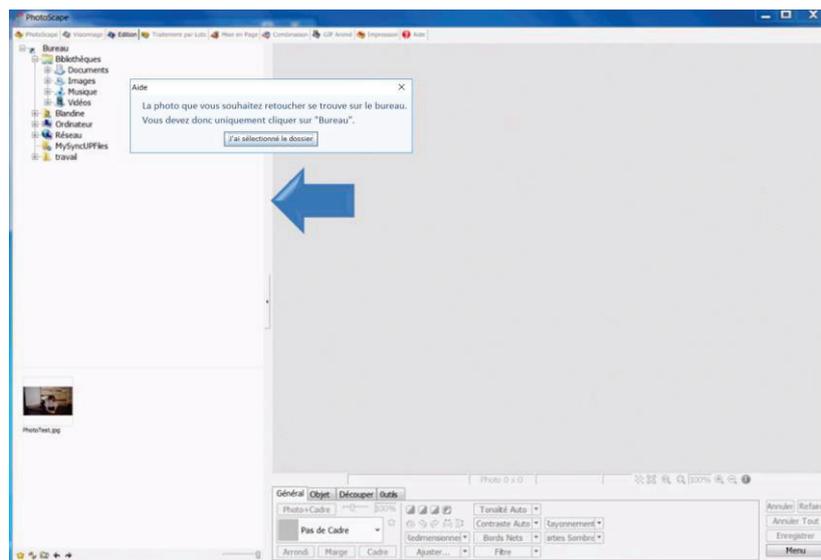


Figure 64 : Exécution de l'étape "demande plus précise de sélection du dossier" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS

R₃ (cf. Figure 63) : La règle R₃ présentée ci-dessus permet d'ajouter une flèche à côté de l'arborescence des dossiers. Dans R₃, lors du choix « plus d'aide » par l'utilisateur, R₃ effectue l'action d'assistance A₂ qui ajoute la flèche puis lance la règle R₄.

R₄ (cf. Figure 65) : La règle R₄ permet de demander à l'utilisateur de sélectionner le dossier avec un texte plus précis. Cette demande est effectuée *via* la consultation utilisateur C₃ dans R₄. Cette étape ne propose pas « plus d'aide » comme l'étape précédente. C₃ contient donc un seul choix qui confirme la sélection de l'utilisateur. Suite à la consultation C₃, la règle R₄ lance l'action A₁ et la règle R₅ pour effectuer l'étape 4b.

C₃ : type de consultation = « consultation utilisateur », texte = « La photo que vous souhaitez retoucher se trouve sur le bureau. Vous devez donc uniquement cliquer sur "Bureau". », choix = « j'ai sélectionné le dossier »

A₁ : type d'action « mise en valeur », type de mise en valeur = « entourer », composant = « 220 » (liste de photos)

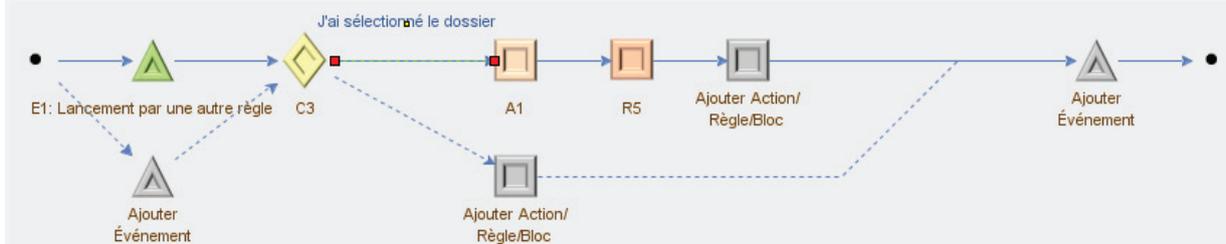


Figure 65 : Définition de la règle R₄ avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

- Étape 4b (cf. Figure 66) : lors de la sélection d'un dossier par l'utilisateur directement dans l'étape 3 ou suite aux aides supplémentaires de l'étape 4a, le système entoure la liste de photos contenues dans le dossier sélectionné, demande d'y sélectionner la photo à corriger et lui demande s'il veut plus d'aide. S'il veut plus d'aide, le système effectue l'étape 5a, sinon, le système effectue l'étape 5b.

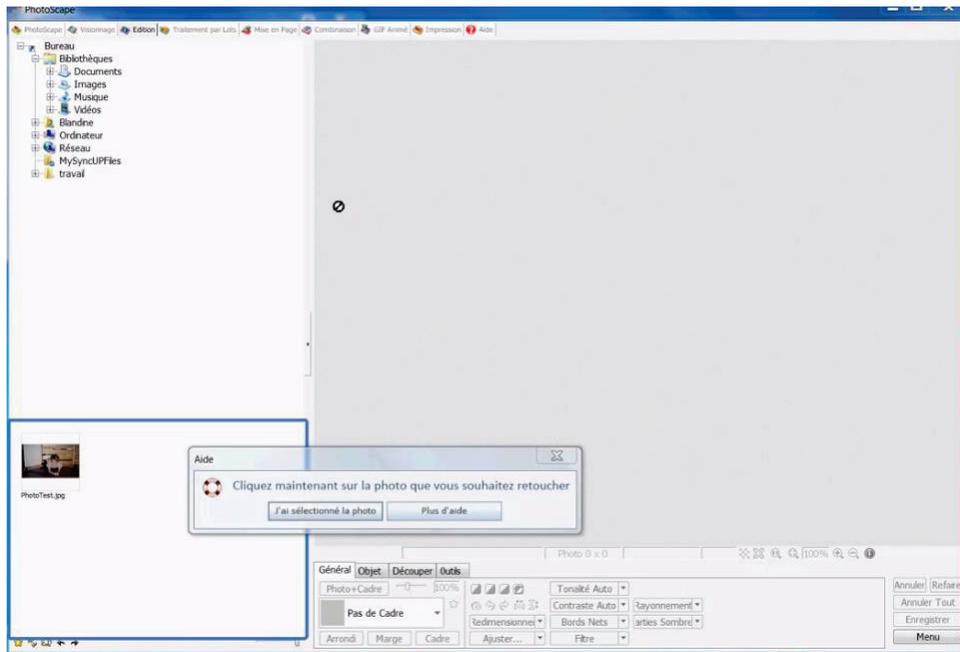


Figure 66 : Exécution de l'étape "demande de sélection d'une photo" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS :

R₃ (cf. Figure 63), R₄ (cf. Figure 65) : Les règles R₃ et R₄ (en cas de demande de plus d'aide) présentées ci-dessus permettent d'entourer l'arborescence des dossiers. Dans R₃ et R₄, lors du choix « j'ai sélectionné le dossier » par l'utilisateur, R₃ et R₄ effectuent l'action d'assistance A₁ qui entoure l'arborescence des dossiers puis lancent la règle R₅.

R₅ (cf. Figure 67) : La règle R₅ permet de demander à l'utilisateur de sélectionner la photo et lui propose plus d'aide. Ces demandes sont effectuées *via* la consultation utilisateur C₄ dans R₅. En fonction de la réponse de l'utilisateur, soit les actions A₃ et A₄ sont lancées pour effectuer l'étape suivante 5a, soit la règle R₆ est lancée pour effectuer l'étape 5b.

C₄ : type de consultation = « consultation utilisateur », texte = « Cliquez maintenant sur la photo que vous souhaitez retoucher », choix = « j'ai sélectionné la photo / plus d'aide »

A₃ : type d'action = « mise en valeur », type de mise en valeur = « afficher un symbole », symbole = « flèche », direction = « par la droite », composant = « 228 » (onglet « édition »)

A₄ : type d'action = « message », texte = « Affichez l'onglet "outils" en cliquant dessus. »

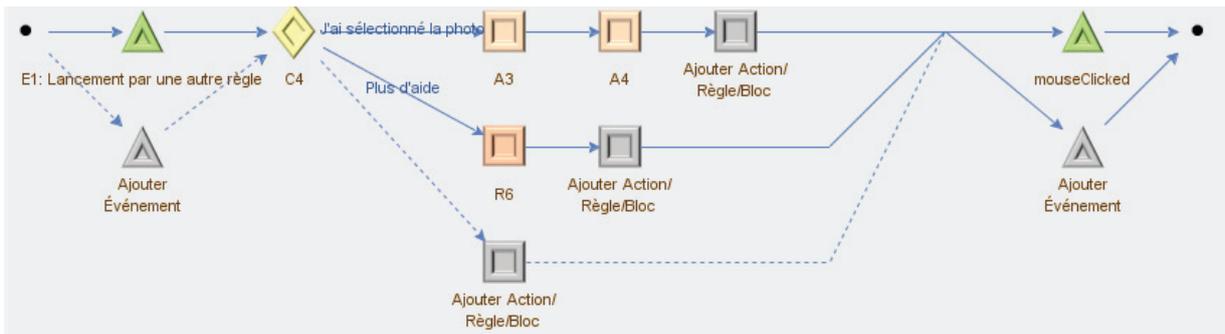


Figure 67 : Définition de la règle R_5 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

- Étape 5a (cf. Figure 68) : lors de la demande d'aide supplémentaire, le système demande également à l'utilisateur de sélectionner la photo dans la liste de photos avec un texte plus précis.

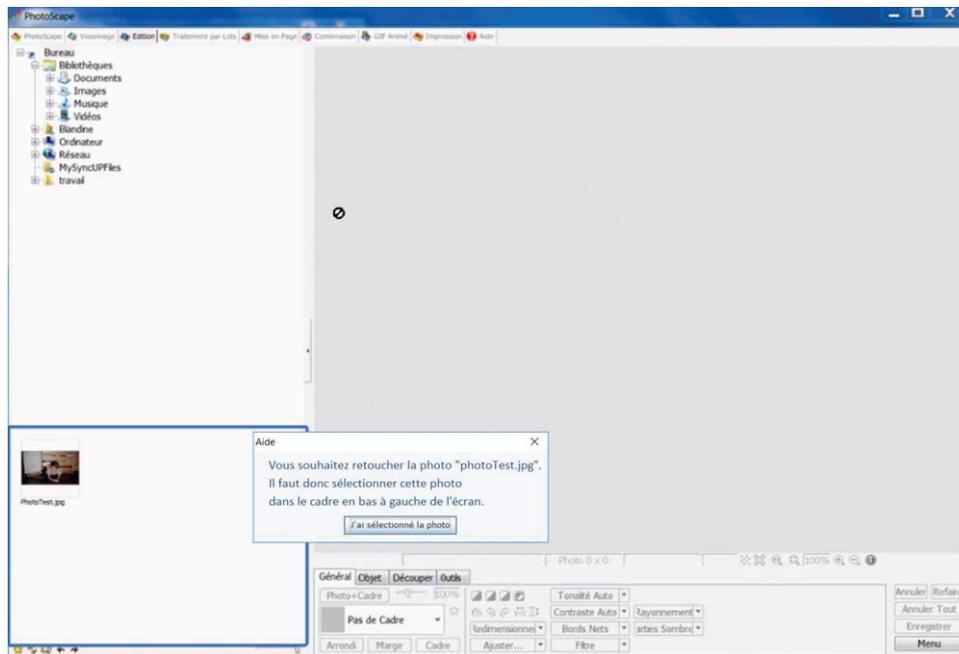


Figure 68 : Exécution de l'étape "demande plus précise de sélection d'une photo" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS :

R_6 (cf. Figure 69) : R_6 permet de demander à l'utilisateur de sélectionner la photo avec un contenu de demande plus précis. Cette demande est effectuée *via* la consultation utilisateur C_5 dans R_6 . Cette règle lance les actions A_3 et A_4 pour effectuer l'étape 5b.

C_3 : type de consultation = « consultation utilisateur », texte = « Vous souhaitez retoucher la photo "photoTest.jpg". Il faut donc sélectionner cette photo dans le cadre en bas à gauche de l'écran. », choix = « j'ai sélectionné la photo »

A_3 : type d'action = « mise en valeur », type de mise en valeur = « afficher un symbole », symbole = « flèche » direction = « par la droite », composant = « 228 » (onglet « édition »)

A_4 : type d'action = « message », texte = « Affichez l'onglet "outils" en cliquant dessus. »

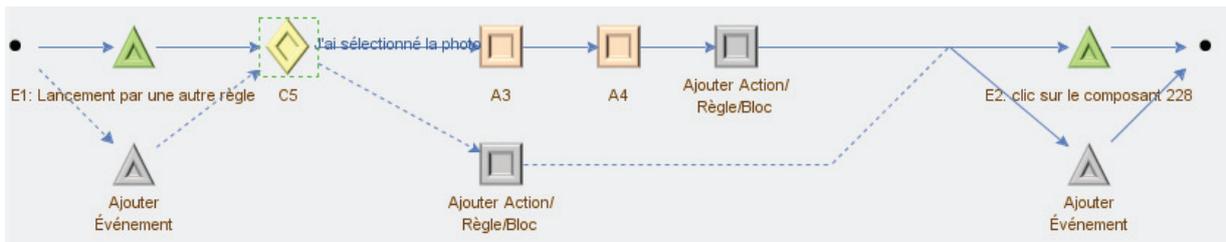


Figure 69 : Définition de la règle R_6 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

- Étape 5b (cf. Figure 70) : lors de la sélection d'une photo, le système ajoute une flèche au-dessus de l'onglet « outils » où le bouton « yeux rouges » se situe et demande de cliquer sur cet onglet. Lors du clic sur cet onglet, la flèche et le message de demande sont terminés et l'étape 6 est effectuée.

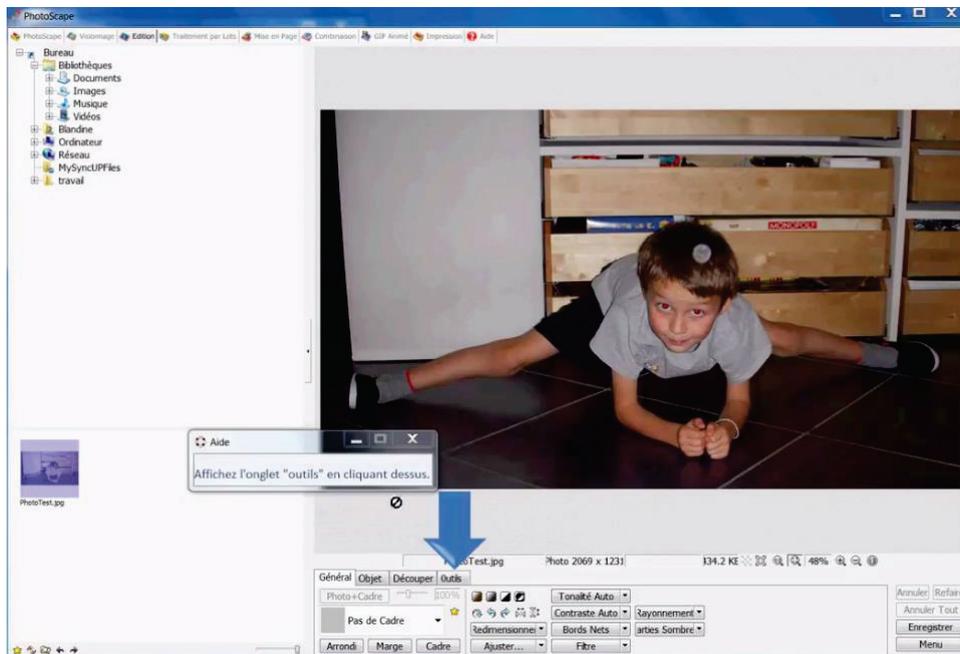


Figure 70 : Exécution de l'étape "demande de clic sur les outils" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS :

R_5 (cf. Figure 67), R_6 (cf. Figure 69) : Les règles R_5 et R_6 (en cas de demande de plus d'aide) présentées ci-dessus permettent d'ajouter une flèche au-dessus de l'onglet « outils » et de demander de cliquer sur cet onglet. Dans R_5 et R_6 , lors du choix « j'ai sélectionné la photo » par l'utilisateur, R_5 et R_6 effectuent l'action d'assistance A_3 qui ajoute une flèche et l'action A_4 qui affiche un message de demande du clic.

- Étape 6 (cf. Figure 71) : lors du clic sur l'onglet « outils », le système ajoute une flèche au-dessus du bouton « yeux rouges » et demande à l'utilisateur de cliquer sur ce bouton. Lors du clic sur ce bouton, l'affichage de la flèche et du message de demande sont stoppés et l'étape 7 est effectuée.

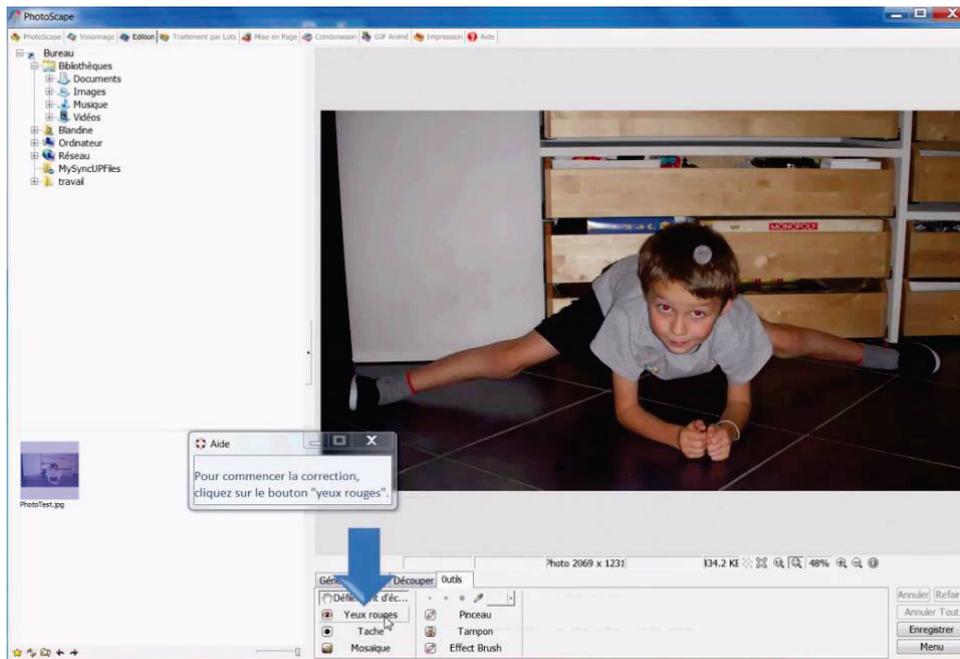


Figure 71 : Exécution de l'étape "demande de clic sur le bouton yeux rouges" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS :

R₇ (cf. Figure 72) : la règle R₇ permet d'ajouter une flèche au-dessus du bouton « yeux rouges ». R₇ débute par l'événement E₂ (lors du clic sur ce bouton). R₇ effectue les actions A₅ et A₆. Cette règle se termine par l'événement E₃ qui termine les actions A₅ et A₆.

E₂ : type d'événement = « action de l'utilisateur », action = « clic », composant = « 228 » (onglet « outils »)

E₃ : type d'événement = « action de l'utilisateur », action = « clic », composant = « 210 » (bouton « yeux rouges »)

A₅ : type d'action = « mise en valeur », type de mise en valeur = « afficher un symbole », symbole = « flèche », direction = « par la droite », composant = « 210 » (bouton « yeux rouges »)

A₆ : type d'action = « message », texte = « Pour commencer la correction, cliquez sur le bouton "yeux rouges". »

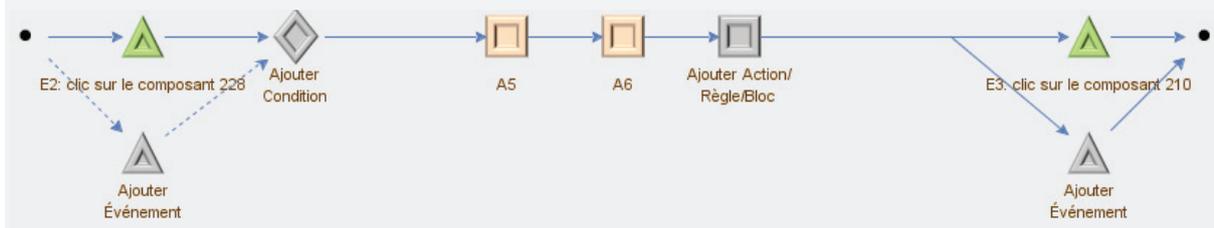


Figure 72 : Définition de la règle R₇ avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

- Étape 7 (cf. Figure 73) : lors du clic sur le bouton « yeux rouges », le système explique à l'utilisateur comment utiliser cette fonctionnalité et lui demande s'il veut plus d'aide. Le cas

échéant, le système affiche une vidéo et refait cette étape. Sinon, le système affiche un message de remerciement qui clôture cette assistance (cf. Figure 75).

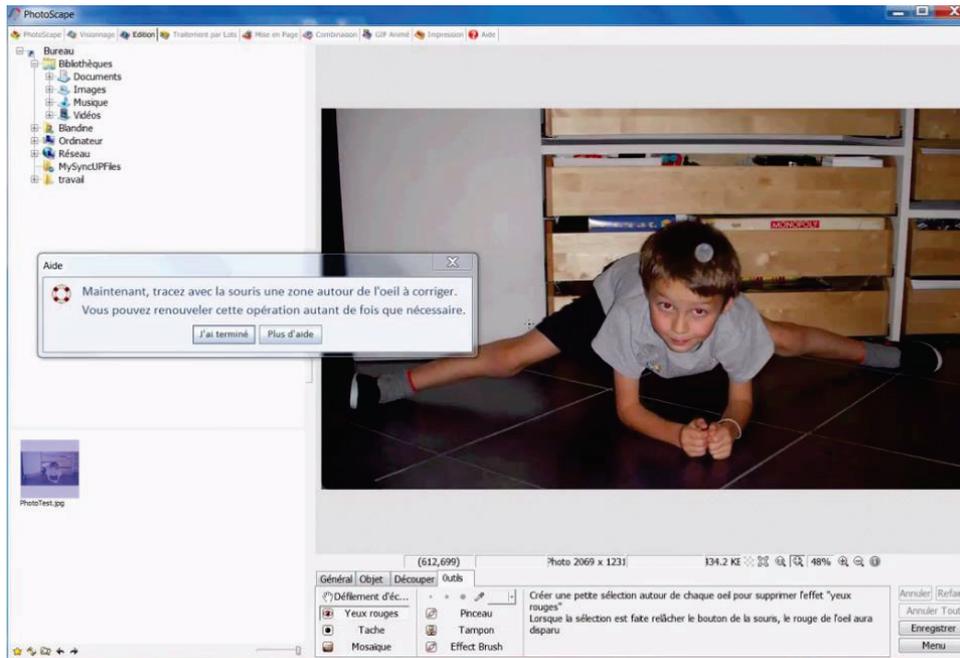


Figure 73 : Exécution de l'étape "explication d'utilisation de la fonctionnalité" du système d'assistance de PhotoScape avec la version bureau du moteur SEPIA

En aLDEAS :

R₈ (cf. Figure 74) : la règle R₈ débute par l'événement E₃. L'explication de l'utilisation de la fonctionnalité et la demande de plus d'aide sont effectuées *via* la consultation utilisateur C₆ dans R₈.

Si l'utilisateur choisit « plus d'aide », R₈ effectue l'action A₇ pour afficher la vidéo et relancer la règle R₈ pour refaire cette étape.

S'il choisit « j'ai terminé », R₈ effectue l'action A₈ pour afficher un message de remerciement.

E₃ : type d'événement = « action de l'utilisateur », action = « clic », composant = « 210 » (bouton « yeux rouges »)

C₆ : type de consultation = « consultation utilisateur », texte = « Maintenant, tracez avec la souris une zone autour de l'œil à corriger. Vous pouvez renouveler cette opération autant de fois que nécessaire. », choix = « j'ai terminé / plus d'aide »

A₇ : type d'action = « ressource externe », action = « ouvrir le fichier », fichier = « demoPhotoScape.avi »

A₈ : type d'action = « message, texte » = « Cette étude est terminée. Merci de votre participation. »

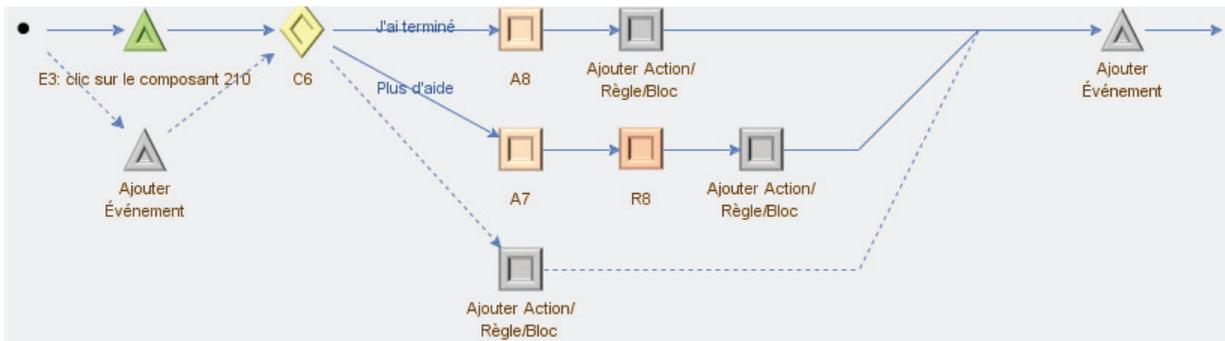


Figure 74 : Définition de la règle R_8 avec l'éditeur SEPIA pour le système d'assistance de PhotoScape

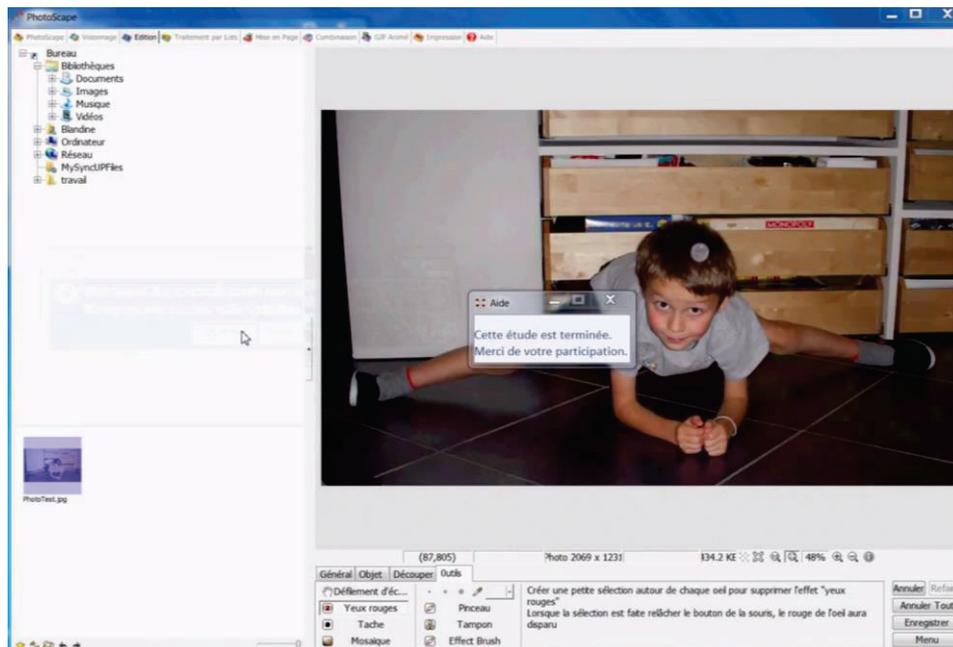


Figure 75 : Affiche du message de remerciement à la fin de l'étape "explication d'utilisation de la fonctionnalité" du système d'assistance de PhotoScape avec le moteur SEPIA

3.6. Discussion du potentiel de SEPIA en contexte éducatif

Le système SEPIA permet aux concepteurs d'assistance de définir des systèmes d'assistance sous forme de règles aLDEAS destinés à compléter une application-cible, et aux utilisateurs de cette application-cible de les exécuter de façon épiphyte sur cette application. Ainsi, SEPIA opérationnalise les deux phases du processus d'adjonction d'un système d'assistance en utilisant le langage aLDEAS comme un langage pivot entre ces deux phases. Le processus et le langage sont génériques. Ils ne sont spécifiques ni à un domaine d'application ni à un type d'application. Cette généricité est pertinente pour l'utilisation de SEPIA en contexte éducatif, en raison de l'existence de domaines variés (mathématiques, informatique, langue, etc.) et de types d'applications très variés (Java, Windows, Web, Linux, MacOS). La création de systèmes

d'assistance contextualisés pour ces types d'applications nécessite l'accessibilité des composants de leurs interfaces. (Ginon, 2014) a identifié et listé les événements détectables et les propriétés consultables des différents composants de l'interface d'applications-cibles selon leur type. Cependant, tous les composants ne sont pas toujours accessibles. Par exemple, le contenu des zones d'édition du logiciel de retouche de photos PhotoScape ne sont pas accessibles, empêchant ainsi SEPIA d'observer l'activité de l'utilisateur sur cette zone spécifique. De même, les parties d'une application-cible développées en Flash ou sous forme d'Applet ne sont pas du tout accessibles par le système SEPIA. De plus, dans certaines applications-cibles, certains composants peuvent être générés automatiquement a posteriori de la description de l'interface. Cela complique et alourdit l'étape de description de l'interface, car il faut identifier tous les composants possibles. Enfin, la contextualisation d'un système d'assistance peut nécessiter certaines informations qui ne peuvent pas être atteintes sur les composants de l'interface, telles que des valeurs stockées en mémoire, des données stockées uniquement en interne, et qui ne sont donc pas atteignables par le système SEPIA. Ces limites techniques constituent le cadre d'application du système SEPIA pour la mise en place de systèmes d'assistance contextualisés.

Malgré ces limites techniques, le système SEPIA se montre adapté aux besoins de notre thèse exprimés dans le chapitre 1.

En effet, SEPIA semble adapté aux concepteurs d'assistance non informaticiens car il n'exige pas de compétences en programmation. Les concepteurs peuvent définir des règles aLDEAS qui s'appuient principalement sur les trois éléments principaux du langage. Ainsi, (Ginon, 2014) a commencé par des expérimentations qui ont évalué la spécification de systèmes d'assistance par des étudiants informaticiens. Ils ont facilement réussi à définir des règles d'assistance aLDEAS. (Ginon, 2014) a également testé le système de création d'assistance avec cinq personnes non-informatiennes. Ces personnes ont chacune réussi à spécifier un système d'assistance de 7 à 9 règles. Cependant, spécifier des systèmes d'assistance est souvent complexe en contexte éducatif dû à la richesse de l'assistance souhaitée et cette définition nécessite une maîtrise avancée du système SEPIA et du langage aLDEAS. La spécification d'un système d'assistance avec SEPIA en contexte éducatif semble donc plutôt destinée à des ingénieurs pédagogiques, qui ne sont généralement pas informaticiens, mais qui pourraient être formés à l'utilisation de SEPIA. Les enseignants quant à eux pourront donc plutôt demander la conception des assistances souhaitées aux ingénieurs pédagogiques. Cela n'empêche pas l'exploitation des assistances par les enseignants auprès de leurs apprenants.

SEPIA permet de fournir une assistance très fine. Ainsi, l'assistance peut être contextualisée en prenant en compte l'état de l'application, peut être personnalisée en prenant en compte le profil de l'utilisateur, et peut être progressive en prenant en compte l'historique de l'assistance. De plus, les formes très variées de l'assistance (message, mise en valeur, agents animés, son, etc.) sont également un facteur d'adaptation de l'assistance aux utilisateurs et au contexte. Dans le contexte éducatif, l'adaptation de l'assistance est nécessaire pour prendre en compte la variété des apprenants et la variété des situations. Par exemple, les apprenants en difficulté peuvent recevoir des messages d'aide très détaillés, tandis que les apprenants avec un handicap visuel peuvent recevoir l'assistance sous forme de messages audio ou de messages textuels de taille, de police et de couleurs adaptés aux spécificités de son handicap.

Ces différents points forts nous conduisent à utiliser le langage aLDEAS et le système SEPIA pour nos travaux sur l'assistance à l'utilisateur en contexte éducatif. Cependant, pour pouvoir

prouver qu'ils y ont complètement adaptés, nous avons confronté les besoins d'assistance en contexte éducatif au langage aLDEAS et au système SEPIA. Pour cette confrontation, nous avons identifié deux questions de recherches : comment définir le déroulement d'une assistance pédagogique et comment définir un guidage pédagogique.

3.6.1 Comment définir le déroulement d'une assistance

Le langage aLDEAS et sa mise en œuvre dans SEPIA permettent d'ores et déjà de décrire le déroulement d'une assistance. aLDEAS permet de définir cinq modes d'articulation entre les éléments d'assistance : indépendant, successif, simultané, progressif et interactif.

Un élément d'assistance peut être un message, une mise en valeur, etc., mais également une combinaison de ces éléments. Dans SEPIA, un élément d'assistance peut être défini par une règle aLDEAS. Ainsi, une règle peut lancer une ou plusieurs actions élémentaires d'assistance ou d'autres règles.

L'articulation entre les éléments d'assistance est donc représentée par l'articulation entre les règles aLDEAS. En général, les règles sont articulées selon un mode indépendant car chacune est lancée par ses propres événements déclencheurs. Cependant, dans des cas particuliers, les règles comportent certains éléments qui conduisent, explicitement ou non, à d'autres modes d'articulations entre règles. Par exemple, les règles qui sont contraintes par les mêmes événements sont lancées simultanément. Les règles qui commencent par la fin d'autres règles s'articulent successivement.

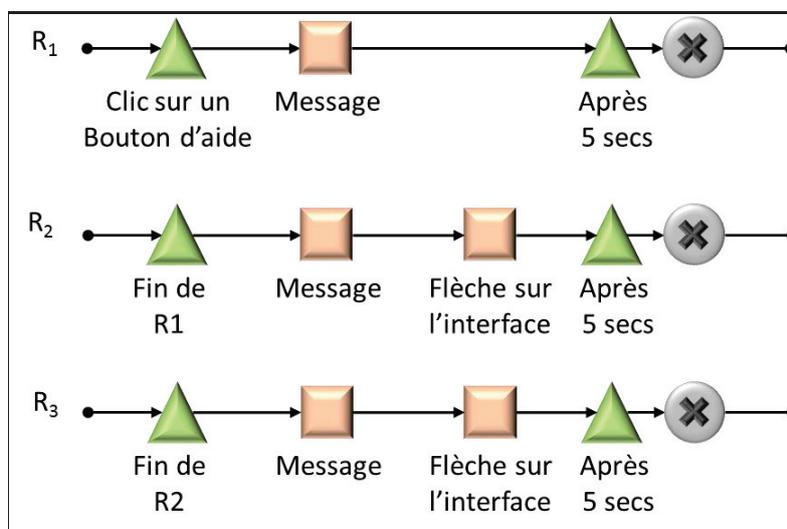


Figure 76 : Exemple de règles aLDEAS permettant une assistance sur ActiveMath selon le mode d'articulation successif

Prenons l'exemple d'une assistance en mode successif qui donne des explications sur ActiveMath. Elle est composée de trois étapes lancées l'une après l'autre. Dans chaque étape, un message et une flèche sur l'interface sont affichés pendant 5 secondes. Pour implémenter cette assistance, il faut trois règles aLDEAS, R_1 , R_2 , R_3 (cf. Figure 76), correspondant à ces trois étapes. Chaque règle prend fin avec un événement de fin « après 5 secondes ». La règle R_1 commence par le clic de l'apprenant sur le bouton d'aide. Elle affiche un message et le ferme après 5 secondes. Pour que la deuxième étape s'enchaîne après la première, la règle R_2 doit

commencer par la fin de la règle R_1 . Donc, à la fin de R_1 , R_2 est lancée. La règle R_2 affiche un message et une flèche pour attirer l'attention de l'apprenant. Le fonctionnement est le même pour la troisième étape implémentée par la règle R_3 .

Nous voyons sur cet exemple que les concepteurs d'assistance peuvent définir une articulation entre des règles aLDEAS. Cependant, l'expression de l'articulation est implicite en aLDEAS et elle peut être complexe à définir pour le concepteur de l'assistance dans SEPIA.

Dans SEPIA, l'assistance est définie par un ensemble de règles aLDEAS de même niveau, représentées à l'interface dans un tableau (cf. Figure 77). L'événement déclencheur, l'événement de fin et la condition de déclenchement sont des éléments centraux pour définir l'articulation entre règles. Cependant, aucun moyen en dehors de ces éléments ne permet d'explicitement cette articulation. Le concepteur doit examiner chaque règle pour connaître l'articulation entre elles. En plus de se concentrer sur le contenu de chaque règle, le concepteur d'assistance doit définir rigoureusement les événements et les conditions afin qu'ils assurent une articulation correcte entre les règles. En particulier, la définition du mode successif est complexe à mettre en place avec les règles aLDEAS. Ainsi, SEPIA s'assure de la conformité des règles définies par rapport au patron de règles, mais ne s'assure pas que l'articulation est correcte. Le concepteur d'assistance peut valider une règle avec n'importe quel événement déclencheur ou n'importe quel événement de fin, n'importe quelle condition, n'importe quelle règle, n'importe quelle action, dès lors que cette règle respecte le patron de règles. Évidemment, SEPIA n'automatise ni la modification, ni la génération des éléments aLDEAS pour assurer une articulation correcte correspondant au souhait du concepteur.

Nom	Évènement déclencheur	Condition	Action	Évènement de fin
R9 : ajout butto...	Lancement de l'assistance		A11	
R15 : organisati...	E5 : mouseClicked sur le composant C0		A18	E11 : 5 secondes
R16 : explicatio...	E12 : fin sur le composant R15		A20, A22	E11 : 5 secondes
R17 : explicatio...	E13 : fin sur le composant R16		A19, A21	E11 : 5 secondes

Figure 77 : Tableau contenant la liste des règles aLDEAS dans SEPIA

L'absence de moyen d'explicitement et de s'assurer d'une articulation correcte entre les règles aLDEAS rend la définition des articulations par le concepteur de l'assistance implicite et difficile. Il nous faut donc proposer des moyens pour faciliter le travail des concepteurs, c'est-à-dire, en contexte éducatif, des enseignants ou des ingénieurs pédagogiques.

3.6.2 Comment définir un guidage pédagogique

Le langage aLDEAS et sa mise en œuvre dans SEPIA permettent déjà de décrire un guidage pédagogique. aLDEAS permet de définir cinq types de guidages pédagogiques : libre, temporel, contextualisé, séquentiel et personnalisé.

Dans SEPIA, un guidage pédagogique est défini par un ensemble de règles. Les règles sont chargées de gérer non seulement l'assistance à la réalisation d'une activité par l'utilisateur final, mais également l'enchaînement entre les activités. Mettre en place un guidage pédagogique avec SEPIA nécessite la définition d'un grand nombre de règles comportant de nombreux éléments aLDEAS. Ainsi, le concepteur peut par exemple utiliser les consultations de

l'utilisateur pour demander à l'apprenant quelle activité il veut faire. Il s'agit dans ce cas d'un guidage libre dans lequel plusieurs activités sont disponibles. En exploitant le profil de l'utilisateur, le concepteur peut créer un guidage personnalisé pour proposer les activités à chaque apprenant de manière individualisée. En consultant l'état de l'application, le concepteur peut déterminer l'état de réalisation d'une activité pédagogique pour diriger l'apprenant vers d'autres activités adaptées à sa situation (dans ce cas le guidage est séquentiel et contextualisé). De plus, le profil de l'utilisateur permet aux concepteurs de stocker l'état d'une activité pour pouvoir l'exploiter plus tard. Cependant, il manque dans SEPIA la possibilité de consulter le temps, ce qui empêche la définition d'un guidage temporel qui détermine la disponibilité d'une activité pour l'apprenant en fonction du moment.

Prenons un exemple de guidage pédagogique contextualisé qui propose à l'apprenant des activités de résolution d'équation. Ce guidage conduit tout d'abord à la première activité. Si l'apprenant y répond correctement, il est dirigé vers une deuxième activité comportant une équation plus difficile, dans le cas contraire il est dirigé vers une troisième activité, une activité de remédiation comportant une équation plus facile. La réalisation de ce guidage nécessite la définition d'un grand nombre de règles. Pour simplifier, nous ne présentons que certaines de ces règles (cf. Figure 78). Lors du lancement de l'assistance, la règle R_{11} permet d'afficher un message suggérant aux apprenants de réaliser la première activité. Lors du clic sur le bouton « terminer » (l'apprenant a terminé son activité), la règle R_{12} permet de vérifier la réponse de l'apprenant. Si celle-ci est correcte, la règle R_{21} est lancée pour demander à l'apprenant de réaliser la deuxième activité. Dans le cas contraire, la règle R_{31} est lancée pour proposer la troisième activité (une activité de remédiation). Les règles R_{11} et R_{12} font partie de la première activité, R_{21} fait partie de la deuxième et R_{31} fait partie de la troisième activité. Plus précisément, R_{12} permet de diriger l'apprenant de la première activité vers l'une des deux autres en fonction de sa réponse. Cette règle permet donc l'organisation des activités dans un guidage contextualisé.

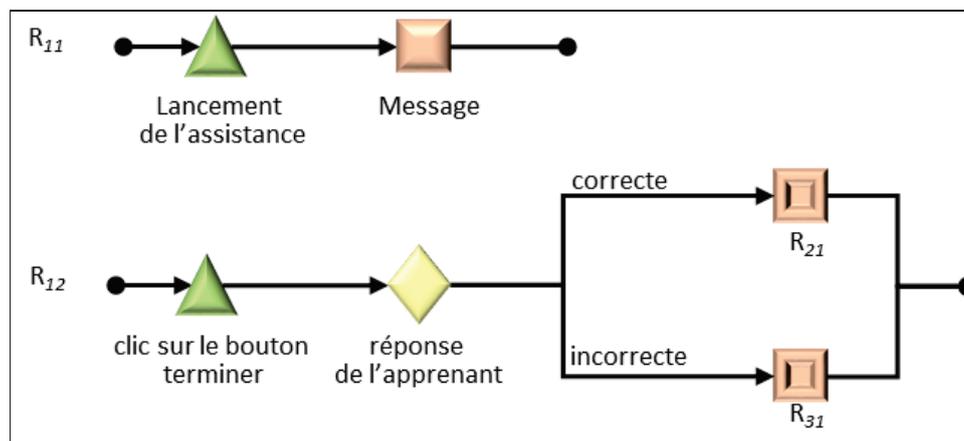


Figure 78 : Extrait des règles définissant un guidage pédagogique contextualisé

Cet exemple montre que les concepteurs d'assistance peuvent définir un guidage pédagogique dans SEPIA à l'aide de règles aLDEAS. Cependant, puisque SEPIA et aLDEAS n'intègrent ni la notion de guidage pédagogique ni celle d'activité d'apprentissage, l'expression d'un tel guidage pédagogique en aLDEAS est implicite et peut être complexe à définir pour le concepteur de l'assistance.

Ainsi, comme le guidage pédagogique est défini par un ensemble de règles aLDEAS, il est difficile pour l'enseignant d'identifier les différentes activités associées au guidage pédagogique et de savoir comment ces activités y sont organisées sous forme de règles. De plus, la spécification de ces règles est également très complexe. L'enseignant doit créer des règles dédiées à l'enchaînement des activités et des règles dédiées à la mise en place d'un guidage pédagogique correct. Si l'enseignant veut conserver les états de sortie des activités pédagogiques réalisées par l'apprenant (par exemple, réussite ou échec, erreurs rencontrées) pour les exploiter ultérieurement, il doit modifier la structure du profil de l'apprenant en y ajoutant les éléments nécessaires. Pour chaque fin d'activité, l'enseignant doit créer les règles qui mettent à jour les états de sortie de cette activité. L'enseignant doit définir rigoureusement l'ensemble de ces règles, tâche qui ne fait pas partie de son activité d'enseignement habituelle. Cette tâche va donc être coûteuse en temps et complexe à mettre en œuvre. Même si elle est déléguée à un ingénieur pédagogique, elle reste néanmoins complexe et longue à réaliser. Comme pour le déroulement d'une assistance, SEPIA n'automatise ni la modification, ni la génération des éléments aLDEAS pour assurer un guidage pédagogique correct correspondant au souhait du concepteur.

3.7. Approche proposée pour adapter aLDEAS et SEPIA au contexte éducatif

Pour pouvoir proposer des modèles et outils qui répondent aux deux questions énoncées précédemment, nous suivons une approche qui s'appuie sur deux objectifs principaux : d'une part proposer une définition explicite et facilitée du déroulement d'une assistance et du guidage pédagogique, d'autre part assurer la compatibilité du moteur d'assistance SEPIA avec les règles d'assistance aLDEAS.

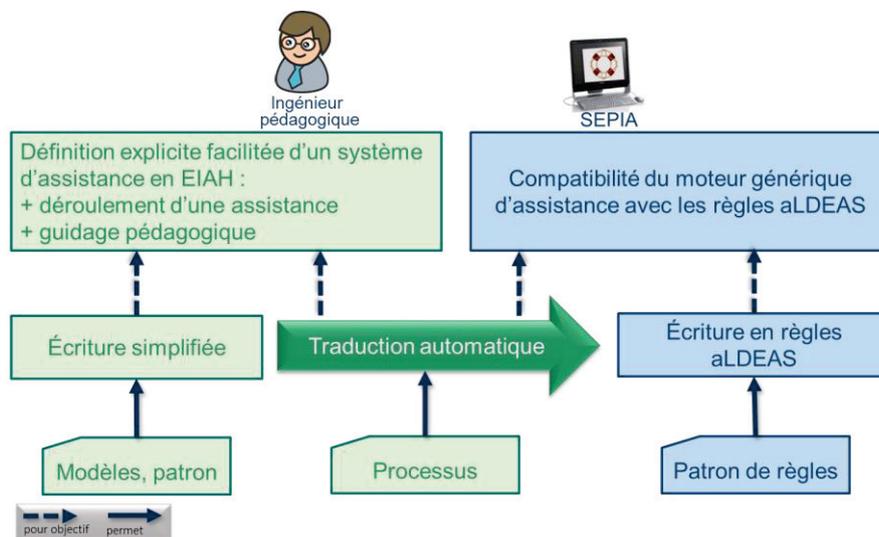


Figure 79 : Approche proposée pour adapter aLDEAS et SEPIA au contexte éducatif

Le premier objectif est lié au contexte éducatif, alors que le second vient du fait qu'un système d'assistance est défini par un ensemble de règles aLDEAS. Le fonctionnement du moteur d'assistance s'appuie sur l'exécution de ces règles aLDEAS. Le maintien de l'écriture

sous forme de règles aLDEAS est donc nécessaire pour garantir le fonctionnement de base du moteur SEPIA. Les règles aLDEAS doivent nécessairement respecter le patron de règles aLDEAS.

Pour expliciter et faciliter la définition d'un système d'assistance en contexte éducatif dans SEPIA, les modèles et patron que nous proposons comme contributions théoriques permettent de proposer une écriture simplifiée à l'ingénieur pédagogique. Cette écriture simplifiée n'est pas comprise par le moteur générique d'assistance SEPIA. Elle doit donc être traduite automatiquement en écriture sous forme de règles aLDEAS. D'une part, la traduction permet d'assurer la compatibilité des règles aLDEAS avec le moteur générique d'assistance SEPIA. D'autre part, cette traduction automatique permet d'alléger la tâche de définition du système d'assistance par l'ingénieur pédagogique. Nous proposons donc des processus qui automatisent une telle traduction.

Dans les deux chapitres suivants, comme contributions de cette thèse, nous proposons des modèles, patron et processus adoptant cette approche. Ces contributions, divisées en deux chapitres, répondent aux deux questions de recherche : « comment définir le déroulement d'une assistance avec SEPIA » et « comment définir un guidage pédagogique avec SEPIA ».

Chapitre 4. Proposition d'articulations entre règles aLDEAS

Objectif de ce chapitre

Proposer de nouveaux modèles et outils permettant d'explicitier et faciliter la définition du déroulement d'une assistance en contexte éducatif avec le langage aLDEAS au sein de SEPIA-edu.

Introduction de ce chapitre

Dans le Chapitre 3, la confrontation du système SEPIA et du langage aLDEAS aux caractéristiques de l'assistance en contexte éducatif nous a amené à deux questions de recherche, dont le traitement est nécessaire au traitement de notre problématique « Comment mettre en place un système d'assistance épiphyte en contexte éducatif en adoptant une approche générique ? » : « Comment définir le déroulement d'une assistance avec SEPIA ? » et « Comment définir un guidage pédagogique avec SEPIA ? ». Dans ce chapitre, nous répondons à la première de ces questions de recherche.

Nous présentons donc notre première contribution : la proposition d'un modèle d'articulation entre règles aLDEAS. Cette proposition vise à rendre explicite la définition de l'articulation entre règles aLDEAS et à faciliter cette définition. Nous proposons donc un **modèle d'articulation entre règles** associé à un **processus de transformation des contraintes en règles aLDEAS** compatibles avec le système SEPIA. Ce modèle et ce processus ont été **implémentés** dans **SEPIA-edu**. L'**exemple** détaillé présenté à la fin du chapitre montre comment définir le déroulement d'une assistance avec SEPIA-edu.

Contributions de ce chapitre

Modèle d'articulation entre règles aLDEAS, modèle de contraintes d'articulation, processus d'application des contraintes aux règles aLDEAS, leur mis en œuvre dans SEPIA

Publications liées à ce chapitre :

Thai, L.-V., Jean-Daubias, S., Lefevre, Ginon, B. : Model of articulation between aLDEAS assistance rules. DC-CSEU (2016).

Thai, L.-V., Jean-Daubias, S., Lefevre, Ginon, B. : Model of articulation between elements of a pedagogical assistance. EC-TEL (2016).

Thai, L.-V., Ginon, B., Jean-Daubias, S., Lefevre, M., Pierre-Antoine, C. : Modèle d'articulation entre les règles définissant un système d'assistance aLDEAS. RJC-IA (2015).

Le déroulement d'une assistance, c'est-à-dire l'articulation entre les différents éléments qui composent cette assistance, est un concept utile dans différents contextes, dont le contexte éducatif. L'assistance peut se dérouler de plusieurs manières différentes (en même temps, étape par étape, etc.). Cependant, SEPIA ne soutient pas suffisamment la définition du déroulement de l'assistance pour aider des enseignants généralement pas informaticiens. La définition de l'articulation entre règles y est implicite et complexe.

Dans ce chapitre, nous présentons donc notre première contribution au projet AGATE, qui vise à répondre à la question « Comment définir le déroulement d'une assistance avec SEPIA ? ». Cette contribution théorique prend la forme d'un modèle d'articulation entre les règles qui définissent l'assistance. Ce modèle comporte cinq modes d'articulations que nous avons identifiés à partir d'une vaste étude bibliographique illustrée dans le Chapitre 2. Le modèle que nous proposons permet une représentation explicite de l'articulation entre règles d'assistance. Nous complétons nos propositions par un processus d'application automatique des contraintes aux règles. Ces contraintes assurent une articulation correcte entre ces règles. Pour illustrer l'utilité de nos modèles, nous déroulons ensuite un exemple étape par étape. Nos modèles sont implémentés dans l'éditeur de SEPIA au travers de deux nouvelles fonctionnalités principales : la définition de l'articulation entre règles d'assistance et l'application semi-automatique des contraintes sur les règles. La définition de l'articulation se fait grâce à une interface graphique, l'articulation entre les règles est ensuite traduite automatiquement par le système en règles aLDEAS compréhensible par le moteur de SEPIA.

4.1. Modèle d'articulation entre règles

Comme nous l'avons montré dans le Chapitre 3, l'articulation entre les différents éléments d'un système d'assistance peut d'ores et déjà être défini à l'aide de règles aLDEAS. Nous avons identifié deux limites à cet existant : la définition de l'articulation entre règles avec SEPIA est à la fois implicite et complexe. Pour remédier à ces limites, nous proposons un modèle d'articulation entre les règles du langage aLDEAS. Ce modèle permet la définition explicite de l'articulation typée entre les différentes règles par le concepteur de l'assistance. Pour regrouper les règles articulées entre-elles, nous ajoutons à aLDEAS la notion de « bloc de règles ». Un bloc de règles est un ensemble de règles articulées dans un mode donné.

La représentation globale de notre modèle d'articulation entre les règles d'un système d'assistance en aLDEAS est donnée en Figure 80. Elle montre une vue d'ensemble des cinq modes d'articulation que nous avons identifiés : indépendant, successif, simultané, progressif et interactif. Dans ce modèle, les règles articulées selon un mode donné sont nommées R_i , avec $i \in [1, n]$ ($n \geq 2$), R_i représentant un élément d'assistance.

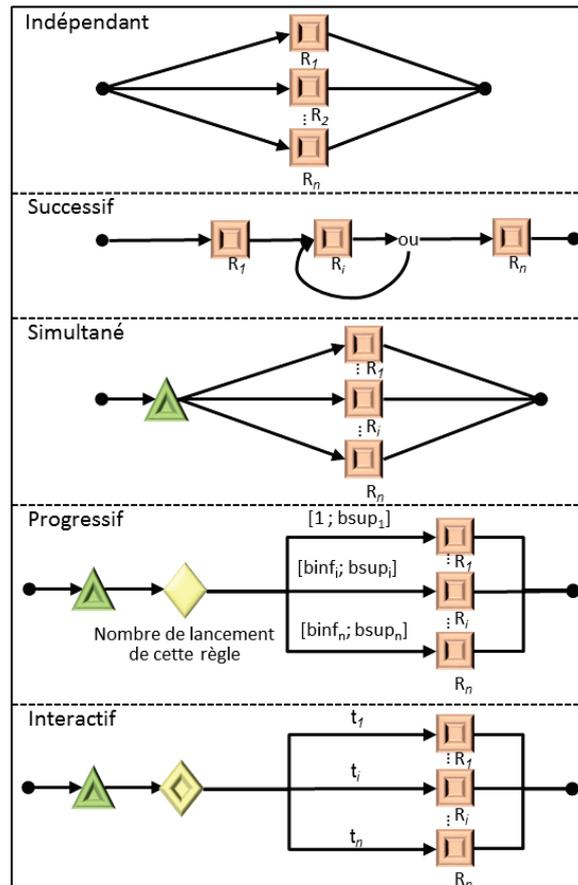


Figure 80 : Modèle d'articulation entre les règles d'un système d'assistance en aLDEAS

Pour chaque mode d'articulation, notre modèle donne sa représentation en aLDEAS. Cette représentation précise les contraintes que doivent respecter les règles. L'articulation est correctement formée si les règles respectent toutes les contraintes du mode d'articulation choisi. Dans la suite de ce chapitre, nous présentons donc en détail chaque mode d'articulation en montrant à quoi correspond ce mode, comment il est représenté en aLDEAS et quelles contraintes il impose sur les règles. Nous terminons chaque description par un retour sur l'état de l'art en modélisant à l'aide de notre modèle des exemples correspondant à chaque type d'articulation. Parce qu'en situation réelle les modes d'articulation entre règles d'assistance ne sont pas toujours utilisés indépendamment, mais peuvent être combinés, nous présentons également notre modèle de combinaison de modes d'articulation.

4.1.1 Mode d'articulation indépendant

Dans le mode indépendant, les règles sont déclenchées indépendamment les unes des autres. Elles débutent par leur propre événement déclencheur. Le mode d'articulation indépendant correspond à une absence d'articulation explicite entre règles, même s'il est toujours possible de trouver un lien, même faible entre elles. Ainsi, ce mode reflète la définition par défaut d'un système d'assistance en règles aLDEAS : les règles sont toutes au même niveau et l'articulation entre elles est implicite.

Dans notre modèle, la représentation d'un bloc de règles dans un mode indépendant doit être conforme à la partie « indépendant » de la Figure 80 : toutes les règles sont au même niveau sans qu'aucun élément aLDEAS (événement déclencheur, condition, etc.) ne forme d'articulation explicite entre elles.

Dans ce mode, il n'y a aucune contrainte spécifique à appliquer aux règles. Ainsi, il suffit que les règles respectent le patron général de règles aLDEAS (cf. Figure 50). C'est pourquoi la définition de ce mode et la définition classique sont identiques.

Un exemple de ce mode indépendant est donné dans IXL Learning (cf. Figure 19). L'assistance consiste à afficher des vues d'ensemble de chacun des cours lorsque l'apprenant passe le curseur dessus. Pour simplifier, nous prenons en compte dans cet exemple uniquement les trois premiers cours proposés par IXL Learning : « A.1 Counting review - 0 to 10 », « A.2 Count to fill a ten frame » et « A.3 Counting review - up to 20 ».

Avec la version initiale du langage aLDEAS, pour définir cette assistance, il fallait créer trois règles R_1 , R_2 , R_3 (cf. partie basse de la Figure 81), chacune se chargeant de l'affichage de la vue d'ensemble d'un des cours. Ces règles débutent par leur propre événement déclencheur « passage de la souris sur le cours », affichent la vue d'ensemble du cours sous forme d'un message et se terminent par la sortie du cours.

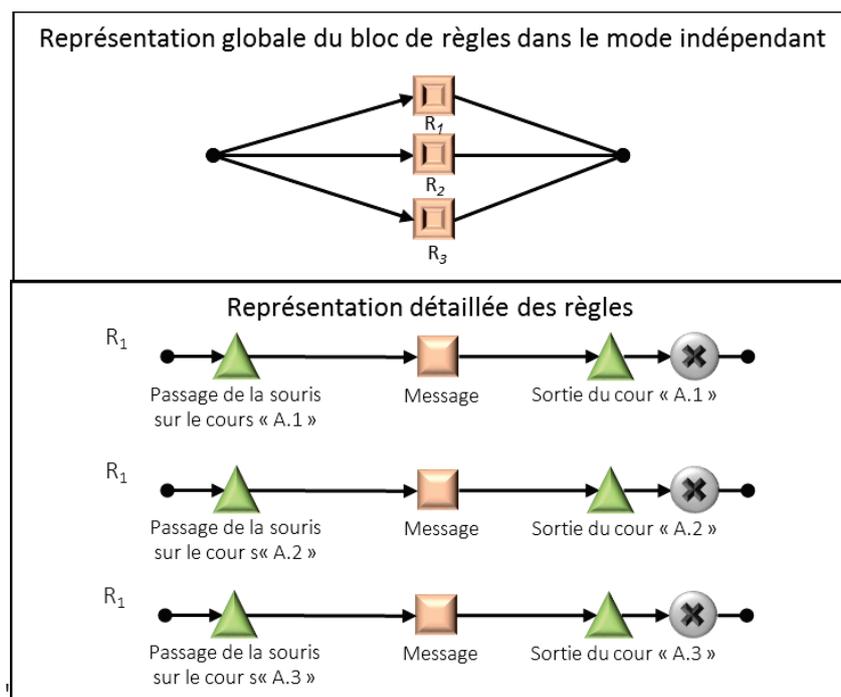


Figure 81 : Exemple d'assistance sur IXL Learning avec trois règles aLDEAS articulées en mode indépendant

Pour définir avec notre modèle d'articulation cet exemple d'assistance d'IXL Learning, nous utilisons un bloc de trois règles R_1 , R_2 , R_3 articulées en mode indépendant. L'articulation entre ces trois règles est illustrée dans la partie haute de la Figure 81 conformément à la représentation du mode indépendant de notre modèle d'articulation entre les règles d'assistance. Comme aucune contrainte spécifique n'est imposée sur les règles, les règles R_1 ,

R_2 , R_3 sont équivalentes à celles définies avec la version initiale du langage aLDEAS (cf. partie basse de la Figure 81).

4.1.2 Mode d'articulation successif

Dans le mode d'articulation successif, les règles sont déclenchées les unes après les autres, c'est-à-dire qu'à la fin d'une règle R_i , la règle R_{i+1} est lancée. Le mode d'articulation successif correspond à des assistances données étape par étape.

Dans notre modèle, la représentation d'un bloc de règles en mode successif doit être conforme à la partie « successif » de la Figure 80. Les règles sont placées dans l'ordre de lancement des différentes parties de l'assistance.

Pour former cet ordre de lancement des règles, les règles doivent respecter les contraintes spécifiques représentées dans la Figure 82. Ainsi, une règle R_{i+1} est lancée à la fin de la règle R_i . Donc, la règle R_i est contrainte à posséder un événement de fin pour pouvoir prendre fin et la règle R_{i+1} à posséder comme événement déclencheur « fin de R_i ». Cette contrainte est valable pour toutes les règles, exceptées la première et la dernière : la première règle R_1 peut débuter par n'importe quel(s) événement(s) déclencheur(s) et la dernière règle R_n peut se terminer par aucun, un ou plusieurs événements de fin. Dans ce mode d'articulation, la règle R_1 est un point d'entrée pour lancer l'ensemble des règles R_i . Les événements déclencheurs de cette règle R_1 sont donc également ceux qui lancent ce bloc de règles articulées en mode successif.

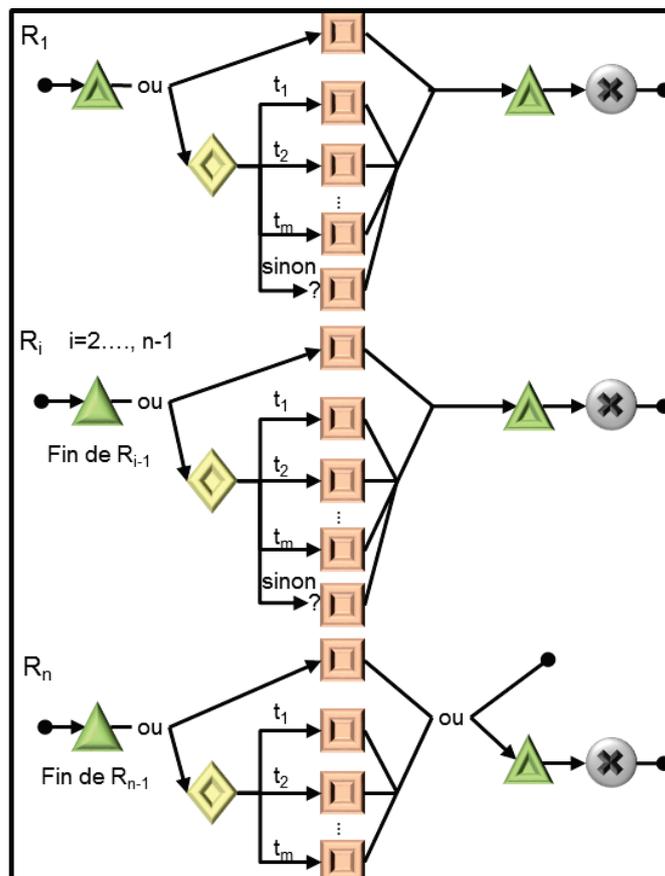


Figure 82 – Représentation en aLDEAS de n règles en mode d'articulation successif

Comme une règle R_{i+1} ne peut être lancée qu'après la fin de la règle R_i qui la précède, si R_i possède une condition de déclenchement qui n'est pas validée au moment de l'exécution de l'assistance, la règle ne sera pas exécutée jusqu'à son événement de fin, et la règle suivante ne sera donc pas déclenchée. Ainsi, en mode successif, si une condition de déclenchement n'est pas validée, toute la succession des règles est interrompue. Cela contraint une règle R_i à avoir une condition possédant une alternative « sinon » (cf. Figure 82). Cette alternative assure que la condition est toujours validée.

Un exemple de ce mode successif est proposé dans le tutoriel de Connectify (cf. Figure 20). Pour simplifier, nous ne prenons en compte que les trois premières étapes du tutoriel. L'assistance définie doit fournir le tutoriel quand l'utilisateur clique sur le bouton « tutoriel ». Ce tutoriel commence systématiquement par la première étape : l'affichage du message « Welcome to Connectify! This tutorial will guide you through setting up a simple Wi-fi Hotspot to share an Internet connection with other devices. ». Cette étape est terminée lors du clic sur le bouton « suivant ». Ensuite, la deuxième étape est lancée pour afficher le message « First, make sure your Internet connection sharing mode is set to create 'Wi-Fi Hotspot' ». Cette étape se termine également lors du clic sur le bouton « suivant ». Enfin, la dernière étape est alors lancée pour afficher le message « Next, choose the Internet connection that you want to share with other devices. Connectify even lets you share the Wi-Fi network that you're currently connected to. Choose 'Help me pick' if you're not sure what to use. ». Ce message est associé à la mise en valeur de la liste déroulante contenant les connexions internet disponibles. La mise en valeur consiste à entourer la liste déroulante. Cette étape est terminée lors du clic sur le bouton « terminer ».

Avec la version initiale du langage aLDEAS, pour définir cette assistance, il fallait créer trois règles (cf. partie basse de la Figure 83) chargées de l'affichage du tutoriel étape par étape. Ces trois règles sont représentées en détail dans la partie basse de la Figure 83. La première règle R_1 débute par l'événement déclencheur « clic sur le bouton tutoriel » et se termine par l'événement de fin « clic sur le bouton suivant ». Lors de ce clic sur le bouton « tutoriel », R_1 affiche un premier message. La règle R_1 et l'affichage du message prennent fin lors de l'événement de fin « clic sur le bouton suivant ». La deuxième règle R_2 , qui suit R_1 , possède l'événement déclencheur « fin de R_1 » et se termine par l'événement de fin « clic sur le bouton suivant ». R_2 affiche le deuxième message. La règle R_2 et son message prennent fin lors du clic sur le bouton « suivant ». La règle R_3 , qui suit R_2 , débute par l'événement déclencheur « fin de R_2 » et se termine avec l'événement de fin « clic sur le bouton terminer ». À la fin de R_2 , R_3 affiche le troisième message et entoure la liste déroulante dont l'identifiant est 330.

Pour définir l'exemple d'assistance de Connectify avec notre modèle d'articulation entre règles, nous utilisons un bloc de trois règles R_1 , R_2 , R_3 articulées en mode successif. L'articulation entre ces trois règles est illustrée dans la partie haute de la Figure 83 conformément à la représentation du mode successif de notre modèle d'articulation entre les règles d'assistance. Les règles articulées R_1 , R_2 , R_3 doivent respecter les contraintes du mode successif. Elles sont équivalentes à celles définies avec la version initiale du langage aLDEAS (cf. partie basse de la Figure 83).

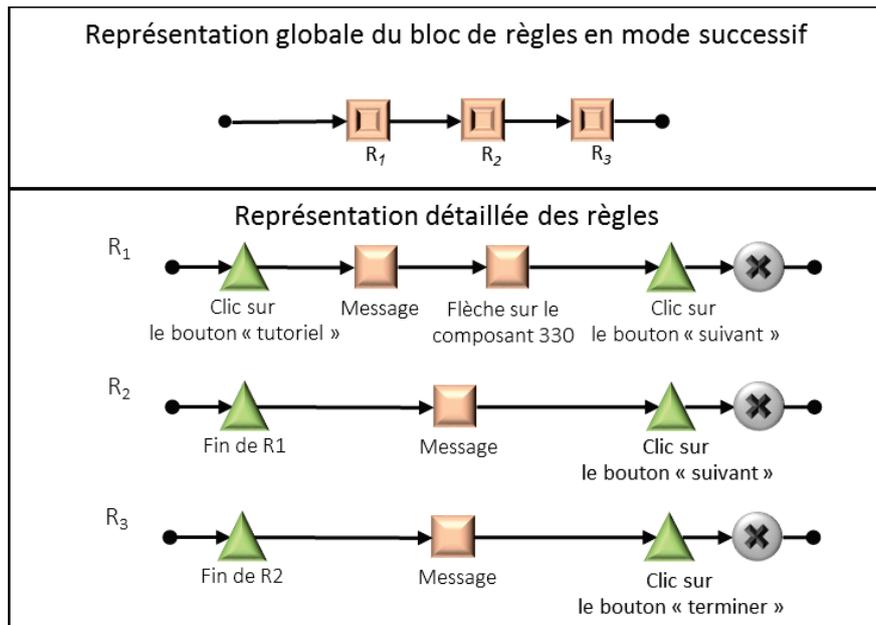


Figure 83 : Exemple du tutoriel de Connectify défini avec trois règles aLDEAS articulées en mode successif

4.1.3 Mode d'articulation simultané

Le mode d'articulation simultané (cf. Figure 80) permet de lancer plusieurs règles d'assistance en même temps, comme c'est le cas avec l'assistance qui met en valeur simultanément toutes les erreurs de saisie de l'utilisateur.

Dans notre modèle, la représentation explicite de l'articulation simultanée doit être conforme à la partie « mode simultané » de la Figure 80. Cette articulation débute par les événements déclencheurs qui la lancent. Toutes les règles articulées R_i sont lancées en même temps.

Dans le mode simultané, les contraintes imposées aux règles sont représentées sur la Figure 84. Les règles R_i doivent respecter ces contraintes. De plus, dans le mode d'articulation *successif*, la règle R_1 est le point d'entrée du système d'assistance pour lancer successivement les différentes règles R_i , alors que dans ce mode d'articulation *simultané*, il faut une règle supplémentaire comme point d'entrée, afin de lancer toutes les règles R_i simultanément (cf. Figure 84). Nous appelons cette règle R' . R' doit comporter un ou plusieurs événements déclencheurs quelconques, et les règles R_i doivent débiter par l'événement déclencheur « lancement par une règle (R') ».

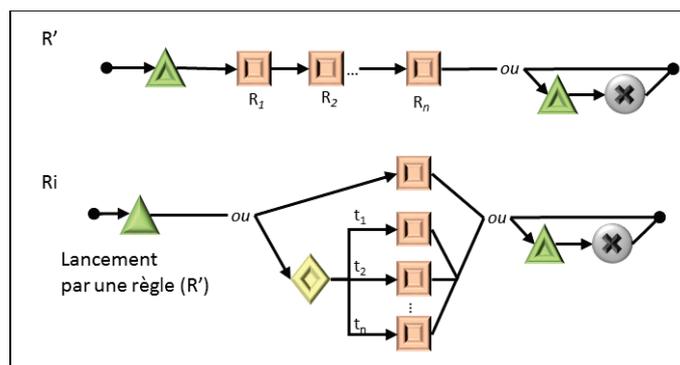


Figure 84 : Représentation en aLDEAS de $n+1$ règles en mode d'articulation simultanée

Un exemple de ce mode d'articulation successif est donné dans le diagnostic des questions des QCM proposés par EOLF (cf. partie droite de la Figure 20). Pour simplifier, nous ne prenons en compte que les trois premières questions. L'assistance définie doit fournir le diagnostic sur les questions en même temps que l'utilisateur clique sur le bouton de vérification. Le caractère « X » est affiché à côté des réponses incorrectes et le mot « OK » est affiché à côté des réponses correctes. Ces caractères sont affichés pendant 10 secondes.

Avec la version initiale du langage aLDEAS, pour définir cette assistance, il fallait quatre règles : R_1 , R_2 , R_3 (cf. partie basse de la Figure 85) qui se chargent du diagnostic des réponses des apprenants et R' qui débute par l'événement déclencheur « clic sur le bouton de vérification » et lance ces trois règles en même temps. R_1 , R_2 , R_3 débutent donc par l'événement déclencheur « lancement par une règle R' » et lancent, en fonction de la réponse de l'apprenant, l'ajout d'un texte (« X » ou « OK ») à côté du composant contenant cette réponse.

Pour définir l'exemple d'assistance d'EOLF avec notre modèle d'articulation, nous utilisons un bloc de trois règles R_1 , R_2 , R_3 articulées en mode simultané. L'articulation entre ces trois règles est illustrée dans la partie haute de la Figure 85 conformément à la représentation du mode simultané de notre modèle d'articulation entre les règles d'assistance. Le bloc débute par un événement déclencheur « clic sur le bouton vérifier » et contient trois règles en parallèle. Les règles articulées R_1 , R_2 , R_3 et la règle R' qui les lance en même temps doivent respecter les contraintes du mode simultané. Elles sont équivalentes à celles définies avec la version initiale du langage aLDEAS (cf. partie basse de la Figure 85).

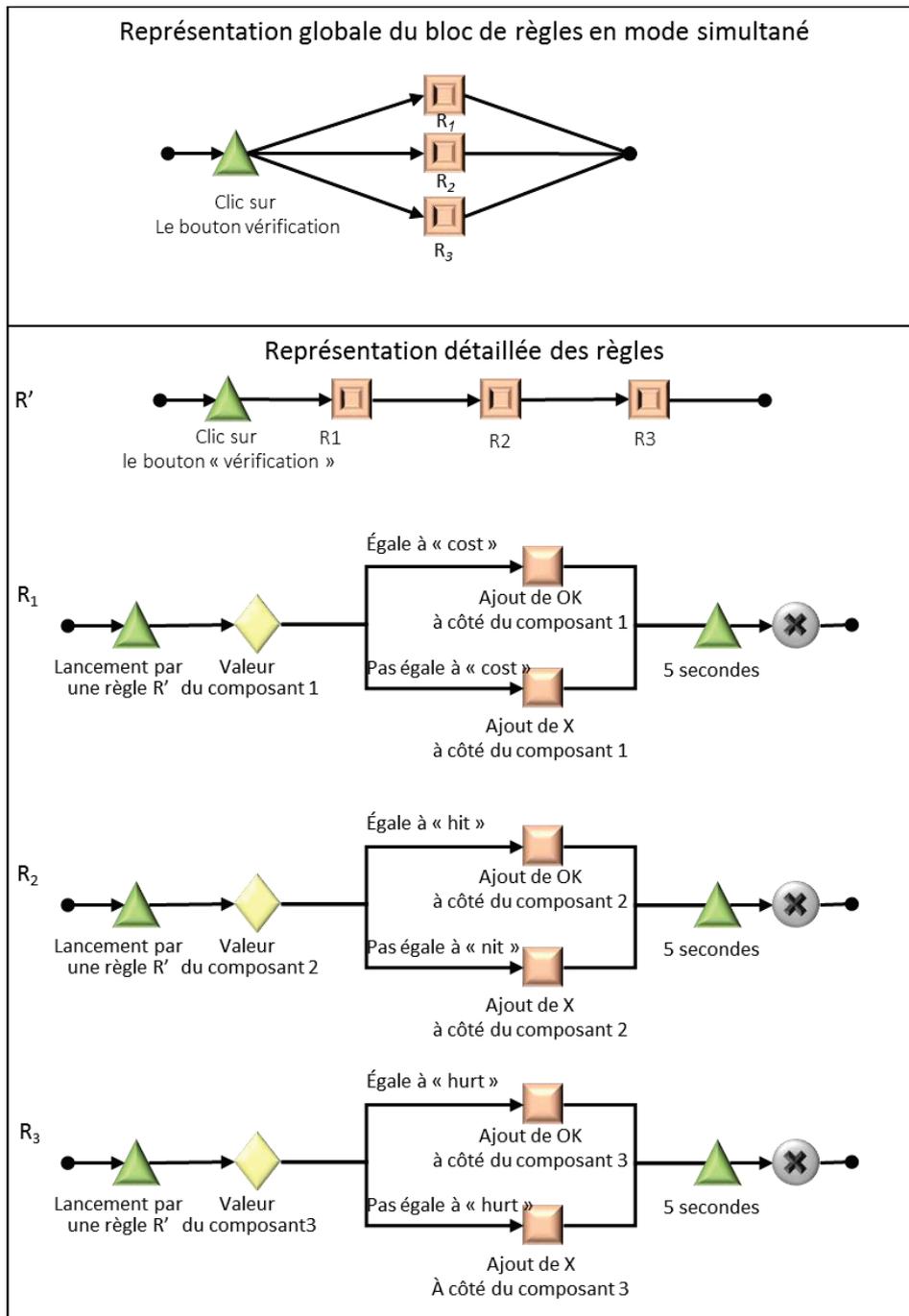


Figure 85 : Exemple du diagnostic des questions sur le site EOLF avec trois règles aLDEAS articulées en mode simultané

4.1.4 Mode d'articulation progressif

Le mode d'articulation *progressif* (cf. Figure 80) permet de déclencher des règles d'assistance différentes selon le nombre de fois où l'utilisateur se retrouve dans une même situation. Ce mode d'articulation permet par exemple de proposer progressivement à l'utilisateur une assistance de plus en plus importante pour répondre à un besoin d'assistance qui se répète.

Dans notre modèle, la représentation explicite de l'articulation progressive doit être conforme à la partie « mode progressif » du modèle d'articulation donné par le Figure 80. Cette articulation débute par les événements déclencheurs qui la lancent. Les règles articulées R_i sont lancées en fonction du nombre de demandes d'assistance, c'est-à-dire du nombre de lancement de cet ensemble de règles. Le nombre de déclenchements d'une règle R_i dépend de l'intervalle $[binf_i, bsup_i]$: R_i sera déclenchée un nombre de fois compris entre la borne inférieure $binf_i$ et la borne supérieure $bsup_i$. Lors de ses $bsup_1$ premiers déclenchements, R_1 est lancée, puis lors de ses déclenchements suivants, c'est R_2 qui est lancée, etc.

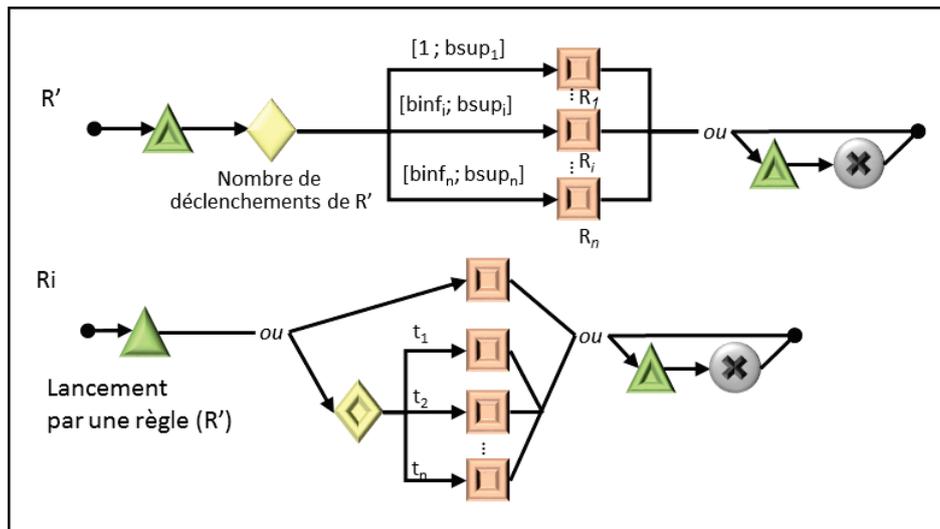


Figure 86 : Représentation en aLDEAS de n règles en mode d'articulation progressif

Dans le mode progressif, les contraintes imposées aux règles sont représentées dans la Figure 86. Les règles R_i doivent respecter ces règles. Plus précisément, pour définir une articulation progressive entre les règles R_i , il faut introduire une règle supplémentaire R' (cf. Figure 86). L'assistance commence par le lancement de cette règle R' , qui lance les règles R_i progressivement. La règle R' débute donc par les mêmes événements déclencheurs que ceux de l'assistance et elle contient une consultation du nombre de fois où la règle R' est lancée. Dans la règle R' , chaque alternative pour le lancement d'une règle R_i est associée avec un intervalle $[binf_i, bsup_i]$. Elle peut se terminer ou non par les événements de fin. Pour être lancées depuis R' , les règles R_i doivent débiter par l'événement déclencheur « lancement par une règle (R') ».

Un exemple de ce mode d'articulation progressif est donné dans l'assistance proposée par ActiveMath (cf. Figure 8). L'assistance définie doit fournir deux indices et une solution de façon progressive quand l'utilisateur demande de l'assistance en cliquant sur le bouton aide. Lors de la première demande, un premier indice est fourni. Lors de la deuxième demande un indice plus détaillé est fourni. Lors de la dernière demande, une solution est fournie pour débloquent l'apprenant en difficulté.

Avec la version initiale du langage aLDEAS, pour définir cette assistance, il fallait créer quatre règles : les trois règles R_1 , R_2 , R_3 (cf. partie basse de la Figure 87) qui se chargent du diagnostic des réponses des apprenants et la règle R' qui lance les règles de façon progressive. La règle R' débute par l'événement déclencheur « clic sur le bouton d'aide », consulte le nombre de déclenchements de R' pour lancer la règle pertinente parmi les trois règles R_1 , R_2 , R_3 . Ces trois

règles débutent donc par l'événement déclencheur « lancement par une règle R' » et lancent les indices ou la solution sous forme de messages.

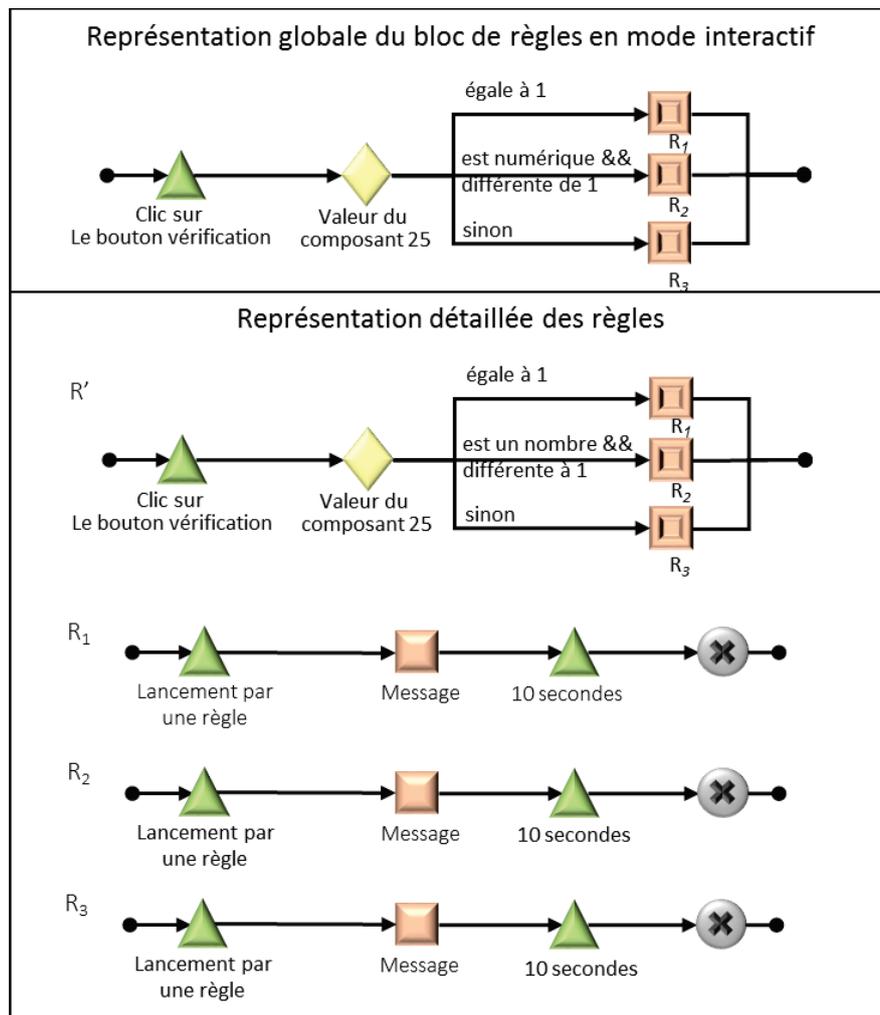


Figure 87 : Exemple d'assistance progressive sur un exercice d'ActiveMath avec trois règles articulées en mode progressif

Pour définir l'exemple d'assistance d'ActiveMath avec notre modèle d'articulation, nous utilisons un bloc de trois règles R_1 , R_2 , R_3 articulées en mode progressif. L'articulation entre ces trois règles est illustrée dans la partie haute de la Figure 87 conformément à la représentation du mode progressif de notre modèle d'articulation entre les règles d'assistance. Le bloc débute par un événement déclencheur « clic sur le bouton d'aide ». Les règles articulées R_1 , R_2 , R_3 et la règle R' qui les lance de façon progressive doivent respecter les contraintes du mode progressif. Elles sont équivalentes à celles définies avec la version initiale du langage aLDEAS (cf. partie basse de la Figure 87).

4.1.5 Mode d'articulation interactif

Dans le mode d'articulation *interactif*, une règle parmi un ensemble de règles R_i est lancée en fonction d'une consultation du profil de l'utilisateur de l'application assistée, de l'état de l'application, de l'historique de l'assistance, des traces et/ou de l'utilisateur. Par exemple, ce

mode d'articulation permet de consulter l'utilisateur pour lui demander de choisir entre plusieurs assistances correspondant à différentes fonctionnalités du logiciel, comme c'est le cas de l'assistance du centre d'aide de Google (Google, 2016). Dans d'autres systèmes, l'assistance proposée à un utilisateur peut différer de celle proposée à d'autres utilisateurs grâce à la consultation du profil de l'utilisateur contenant des informations relatives à ses préférences ou à son expérience.

Dans notre modèle, la représentation explicite de l'articulation de type interactif doit être conforme à la partie « mode interactif » du modèle d'articulation (cf. Figure 80). Cette articulation débute par les événements déclencheurs qui la lancent. Les règles articulées R_i sont lancées en fonction de la consultation. Le lancement d'une règle R_i dépend la valeur t_i associée à chaque alternative de la consultation.

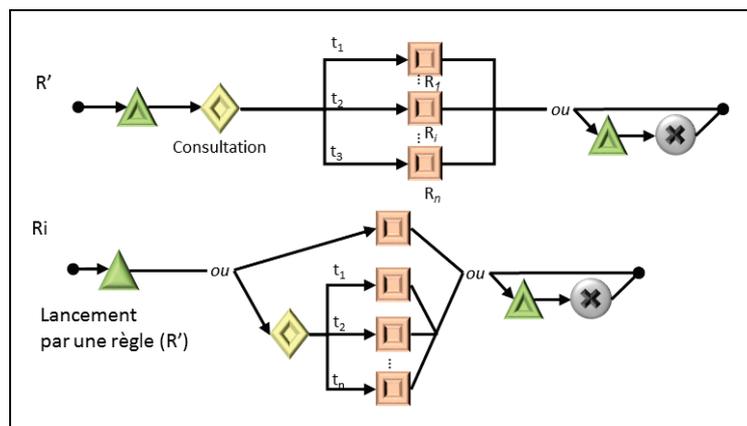


Figure 88 : Représentation en aLDEAS de n règles en mode d'articulation interactif

Dans le mode interactif, les contraintes imposées aux règles sont représentées dans la Figure 88. Plus précisément, pour définir l'articulation interactive entre les règles R_i , il est nécessaire d'introduire une règle supplémentaire R' . L'assistance commence par le lancement de cette règle qui lance les règles R_i progressivement. Elle débute donc par les mêmes événements déclencheurs que ceux de l'articulation. La règle R' contient également une consultation identique à celle continue dans l'articulation. Dans la règle R' , chaque alternative pour le lancement d'une règle R_i est associée à une valeur t_i . Elle peut se terminer ou non par les événements de fin. Pour être lancées depuis R' , les règles R_i doivent débiter par l'événement déclencheur « lancement par une règle (R') ».

Un exemple de ce mode d'articulation interactif est proposé dans le diagnostic de l'exercice d'ActiveMath (cf. Figure 89). L'assistance définie doit fournir le diagnostic sur la réponse de l'apprenant quand celui-ci demande de l'assistance en cliquant sur le bouton « vérification ». Ce diagnostic est établi grâce à la consultation de la réponse de l'apprenant, qui est la valeur contenue dans le champ à saisir. Il y a trois types de réponses : correcte (si la valeur est 1), incorrecte (si la valeur est un nombre différent à 1) et erreur de syntaxe (si la valeur n'est pas un nombre).

Avec la version initiale du langage aLDEAS, pour définir cette assistance, il fallait créer quatre règles (cf. partie basse de la Figure 89) : trois règles R_1 , R_2 , R_3 qui se chargent du diagnostic des réponses des apprenants et une règle R' . La règle R' débute par l'événement déclencheur « clic sur le bouton de vérification », consulte la valeur du composant 25 (la réponse de l'apprenant)

pour lancer la règle adaptée parmi les trois règles R_1 , R_2 , R_3 . Ces trois règles débutent donc par l'événement déclencheur « lancement par une règle R' » et donnent le résultat du diagnostic sous forme d'un message (« réponse correcte », « réponse incorrecte », « erreur de syntaxe »).

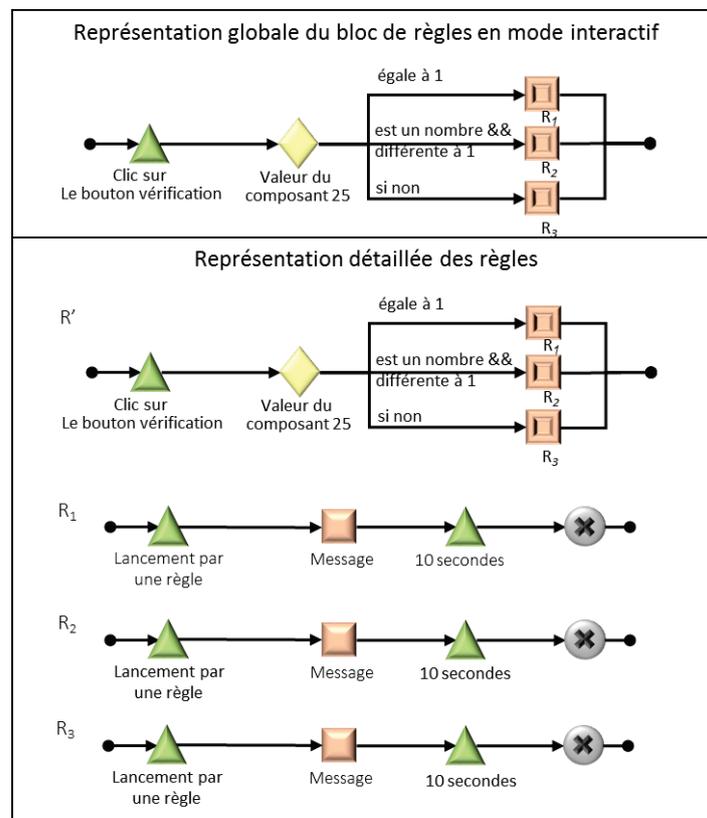


Figure 89 : Exemple du diagnostic sur un exercice d'ActiveMath avec trois règles articulées en mode progressif

Pour définir l'exemple d'assistance d'ActiveMath avec notre modèle d'articulation, nous utilisons un bloc de trois règles R_1 , R_2 , R_3 articulées en mode interactif. L'articulation entre ces trois règles est illustrée dans la partie haute de la Figure 89 conformément à la représentation du mode interactif de notre modèle d'articulation entre les règles d'assistance. Le bloc débute par un événement déclencheur « clic sur le bouton de vérification ». Les règles articulées R_1 , R_2 , R_3 et la règle R' qui les lance doivent respecter les contraintes du mode interactif. Elles sont équivalentes à celles définies avec la version initiale du langage aLDEAS (cf. partie basse de la Figure 89).

4.1.6 Combinaison de modes d'articulation

Un système d'assistance n'utilise pas forcément un seul mode d'articulation entre règles. La combinaison de modes d'articulation se fait grâce à un ou plusieurs blocs intégrés dans un autre bloc. Autrement dit, un bloc peut donc contenir non seulement des règles, mais également des blocs qui sont articulés selon un mode donné. Dans ce cas, R_i est soit une règle, soit un bloc de règles. Cependant, afin d'éviter un niveau inutile, un bloc de règles articulées en mode indépendant n'est pas contenu dans un autre bloc. Ainsi, ce bloc ne peut pas être lancé depuis

d'autres règles ou d'autres blocs car il ne contient pas de règle d'entrée et les règles de ce bloc sont lancées indépendamment.

Prenons un exemple dans lequel il existe une combinaison de plusieurs modes d'assistance sur un exercice de proportionnalité du site Matou Matheux. Dans cet exercice, un élève de seconde doit transformer une vitesse de 15 km/s au format $a \cdot 10^b$ km/h. Il doit donc trouver les valeurs de a et de b. L'assistance que nous avons imaginée lui est fournie de façon progressive. Cette assistance est lancée lors d'une demande de l'élève en difficulté. Lors de la première demande, un exemple similaire à cet exercice est affiché. Cet exemple se compose de trois étapes successives. Pour la deuxième demande, un diagnostic est fourni sur les champs de saisie de a et b en mettant en rouge les champs contenant des valeurs incorrectes.

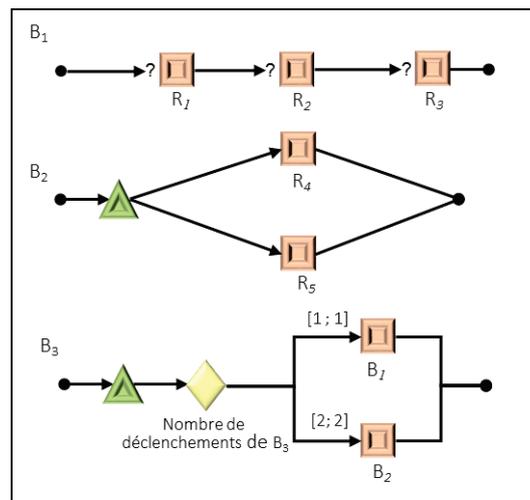


Figure 90 : Représentation de la combinaison de trois blocs de règles

Pour définir cette assistance, nous utilisons notre modèle d'articulation entre règles. Tout d'abord, pour l'exemple donné sous forme de trois étapes successives, il faut un bloc B_1 contenant les trois règles correspondantes articulées en mode successif : R_1 , R_2 , R_3 . La réalisation du diagnostic sur deux champs en même temps nécessite un bloc B_2 de deux règles articulées en mode simultané : R_4 , R_5 . L'exemple et le diagnostic sont fournis progressivement. Il faut donc un bloc B_3 contenant B_1 et B_2 articulés en mode progressif. La représentation des trois blocs conformes au modèle d'articulation entre règles est illustrée dans la Figure 90.

Comme lors de l'utilisation d'un mode d'articulation, l'utilisation de règles R_i et/ou de blocs B_i dans la combinaison de modes d'articulation doit respecter des contraintes. D'une part, les règles R_i / blocs B_i doivent naturellement respecter les contraintes du bloc qui les contiennent. D'autre part, les contraintes sur ces blocs imposent d'autres contraintes sur les règles et/ou blocs qu'ils contiennent. Prenons par exemple, des règles R_i / blocs B_i articulés dans un bloc B en mode progressif. Ce bloc impose, pour chaque bloc B_i , la contrainte « B_i débute par un événement déclencheur E ». Cette contrainte impose ensuite, pour la règle d'entrée de B_i (la règle R' pour le mode simultané, progressif, interactif et la première règle pour le mode successif), la contrainte « cette règle débute par un même événement déclencheur E ». La représentation des contraintes dans la combinaison de modes est trop complexe pour être réalisée avec la représentation en règles aLDEAS utilisable pour un mode unique. Nous présentons donc ces contraintes sous forme d'un algorithme dans le processus d'application automatique des contraintes des modes d'articulation aux règles que nous proposons. Ce

modèle, qui permet d'assurer le respect des contraintes sur les règles, est donné dans la section suivante. Puis, nous présentons le détail de ces règles dans notre processus à travers l'exemple de la combinaison de trois blocs de règles.

4.2. Processus d'application automatique des contraintes des modes d'articulation aux règles

L'application des contraintes des modes d'articulation sur les règles permet l'articulation correcte entre les règles. Les contraintes de chaque mode sont représentées dans le modèle d'articulation entre règles. Cependant, cela n'assure pas que le concepteur d'assistance puisse appliquer correctement les contraintes sur les règles, car il peut ne pas suivre scrupuleusement le modèle d'articulation entre règles. De plus, cette tâche est complexe surtout quand il est question d'établir une combinaison de modes d'articulation.

Il est donc nécessaire d'automatiser l'application des contraintes pour faciliter le travail du concepteur de l'assistance. Cette application des contraintes permet la génération et la modification des éléments aLDEAS, ainsi que la notification au concepteur des contraintes non respectées pour qu'il puisse les corriger. Ainsi, la plupart des contraintes donne lieu à la génération et/ou à la modification d'éléments aLDEAS. Par exemple, une règle R_i dans un bloc en mode progressif doit débiter par un événement déclencheur « lancement par une règle ». Grâce à l'application de cette contrainte, l'événement « lancement par une règle » est automatiquement ajouté comme événement déclencheur de R_i . Cependant, dans le mode successif, la première règle R_1 doit posséder au moins un événement de fin. Cette contrainte ne précise pas quel événement doit être ajouté comme événement de fin dans R_1 . Elle ne donne donc pas lieu à la génération automatique d'un événement de fin, mais à une notification au concepteur d'assistance.

Pour pouvoir appliquer les contraintes sur les règles, il faut identifier ces contraintes. L'identification des contraintes n'est pas triviale, car pour la combinaison de modes d'articulation, les contraintes sur les règles sont imposées par deux sources différentes : le mode du bloc qui les contient et les contraintes sur ce bloc. Nous avons donc proposé un modèle de contraintes qui permet la création et la transformation de ces contraintes.

Pour automatiser l'application des contraintes des modes d'articulation, nous avons aussi proposé un processus qui composé de trois étapes : (1) initialisation des listes des règles, des blocs et des contraintes qui sont ensuite exploitées et complétées par les étapes suivantes, (2) identification des contraintes sur les règles, et (3) application des contraintes sur les règles.

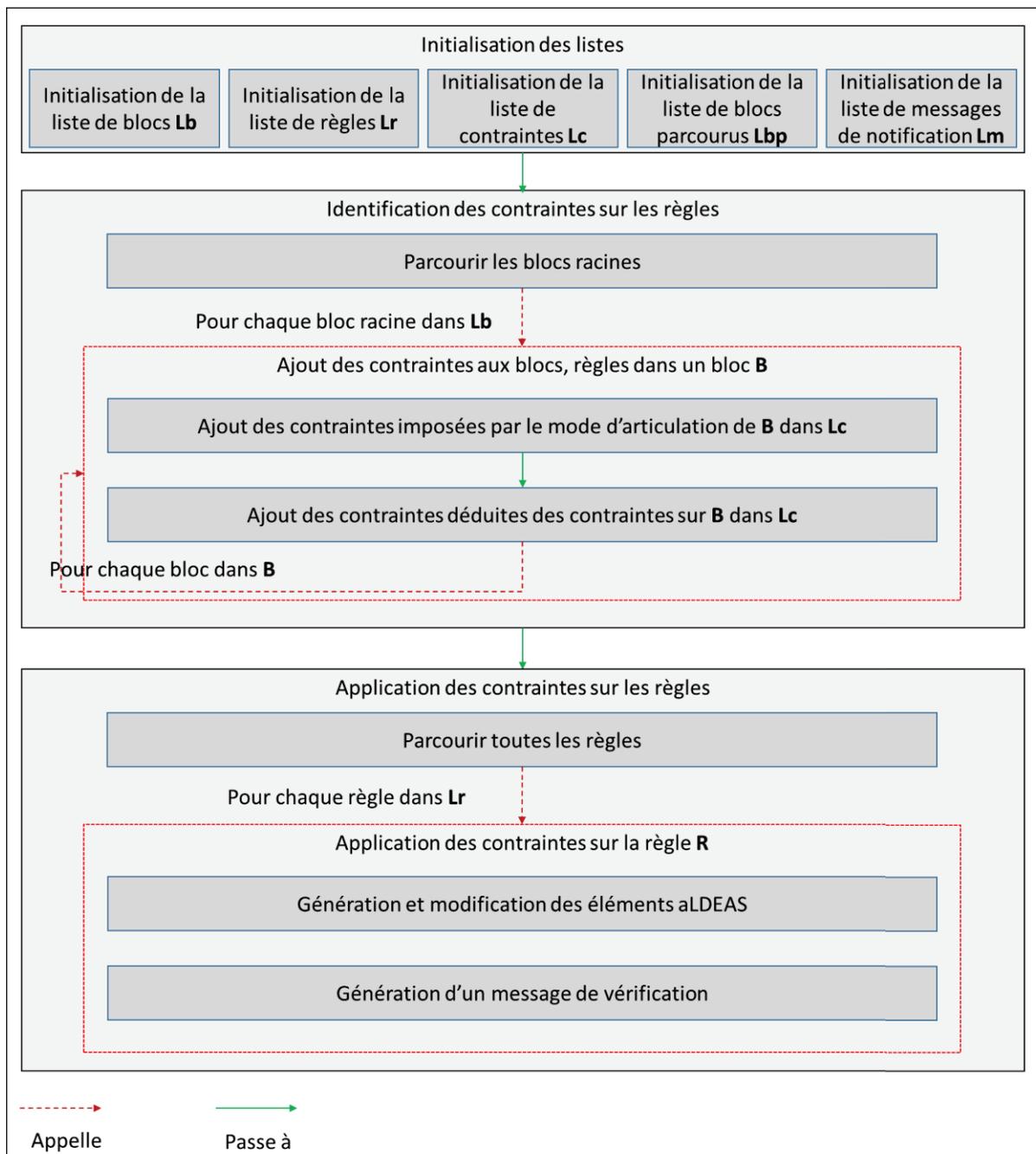


Figure 91 : Processus d'application des contraintes d'articulation

4.2.1 Modèle de contraintes du mode d'articulation sur les règles ou blocs

Le modèle de contraintes est nécessaire pour opérationnaliser le processus d'application des contraintes des modes d'articulation aux règles. Comme nous avons identifié deux sources qui imposent les contraintes sur les règles : le mode du bloc qui les contient et les contraintes sur ce bloc, une contrainte est imposée non seulement sur une règle mais aussi sur un bloc. Les contraintes sur les règles sont très nombreuses. Ici, nous décrivons uniquement les contraintes

qui concernent le modèle d'articulation entre règles car seules ces contraintes peuvent donner lieu à la génération et/ou la modification d'éléments aLDEAS.

Dans notre modèle de contraintes, une contrainte respecte le format Cx(P). Cx est l'identifiant d'un type de contrainte. Le nombre de paramètres P dépend du type de contraintes. Ces types de contraintes sont détaillés dans la Figure 92.

C_creation_regle(R)

Description : une règle R doit être créée.

Modes d'articulation concernés : simultané, progressif, interactif. Ces modes exigent une règle d'entrée R' qui lance le bloc de règles de façon simultanée, progressive ou interactive.

Exemple : C_creation_regle(R') crée une règle R'.

C_lancement_regle(R)

Description : la règle R doit débiter par l'événement « lancement par une règle ».

Modes d'articulation concernés : simultanée, progressif, interactif. Ces modes exigent une règle d'entrée R' qui lance le bloc de règles de façon simultanée, progressive ou interactive.

Exemple : C_lancement_regle(R') ajoute l'événement déclencheur « lancement par une règle » à la règle R'.

C_suite_de_regles(R, R_x)

Description : La règle R doit débiter par l'événement déclencheur « fin de R_x »

Modes d'articulation concernés : successif. Dans ce mode, la règle suivante doit débiter à la fin de la règle précédente.

Exemple : pour C_suite_de_regle(R₂, R₁), R₂ doit débiter par l'événement « fin de R₁ »

C_debut_regle(R, E)

Description : la règle R doit débiter par l'événement déclencheur E

Modes d'articulation concernés : tous. Dans tous les modes, une règle peut être contrainte à débiter par un événement déclencheur : « fin d'une règle précédente », « lancement par une règle », etc.

Exemple : pour C_debut_regle(R₁, E₁), R₁ doit débiter par l'événement déclencheur E₁.

C_fin_regle(R)

Description : La règle R doit prendre fin.

Modes d'articulation concernés : successif. La règle précédente doit prendre fin pour que la règle suivante soit lancée. Cette contrainte exige qu'une règle possède au moins un événement de fin et une alternative « sinon » si elle possède une consultation.

Exemple : pour C_fin_regle(R₁), R₁ doit se terminer par au moins un événement de fin.

C_nb_declenchements(R)

Description : La règle R doit contenir la consultation du nombre de déclenchements de R.

Modes d'articulation concernés : progressif. La règle d'entrée R' est contrainte à posséder la consultation du nombre de déclenchements de R'.

Exemple : pour $C_nb_declenchement(R_1)$, R_1 doit contenir la consultation du nombre de déclenchements de R_1 .

$C_consultation(R, C)$

Description : La règle R doit contenir la consultation C.

Modes d'articulation concernés : interactif. La règle d'entrée R' est contrainte à posséder une consultation qui forme l'articulation interactive.

Exemple : pour $C_consultation(R_1, C)$, R_1 doit contenir la consultation C_1 .

$C_alternative(R, R_x, t_x)$

Description : R doit contenir une alternative avec une valeur t_x . La règle R_x ou la règle d'entrée du bloc R_x est associée à cette alternative.

Modes d'articulation concernés : progressif, interactif. La règle d'entrée R' doit contenir une consultation pour lancer les règles articulées. Les alternatives pour lancer ces règles doivent être ajoutées dans R'.

Exemple : pour $C_alternative(R_1, R_2, 5)$, R_1 doit contenir une alternative avec la valeur 5 associée à la règle R_2 .

Figure 92 : Liste des contraintes liées aux modes d'articulation

4.2.2 Initialisation des listes

L'étape d'initialisation des listes de notre processus d'application automatique des contraintes (cf. Figure 91) permet d'initialiser les listes de blocs, de règles et de contraintes qui sont exploitées et complétées lors des étapes suivantes jusqu'à obtenir, à la fin du processus, un ensemble de règles qui respectent toutes les contraintes d'articulation entre les règles.

Dans cette étape, les blocs de règles articulées selon le mode indépendant ne sont pas pris en compte, car ce mode n'impose aucune contrainte sur les règles. La liste L_b contient donc les blocs de règles articulées selon les modes successif, simultané, interactif et progressif. La liste L_r contient toutes ces règles articulées dans les blocs L_r .

Les contraintes sur un bloc imposent des contraintes sur d'autres blocs et règles, alors que les contraintes sur une règle effectuent la génération et la modification des règles aLDEAS. Il est donc nécessaire qu'une liste L_c contienne les contraintes sur les blocs et sur les règles. Ces contraintes respectent le format « $C_x(P)/R$ ». L_c permet de ce fait de savoir quelles contraintes doivent être appliquées à quels blocs et quelles règles sont nécessaires pour pouvoir appliquer ces contraintes. L'application des contraintes dans L_c peut donner lieu à la génération, à la

modification des règles dans Lr, et/ou à la production des messages de notification des contraintes non respectées dans la liste Lm.

À l'issue de l'initialisation des listes, la liste de contraintes Lc sera complétée lors de l'étape d'identification des contraintes.

4.2.3 Identification des contraintes sur les règles

L'étape d'identification des contraintes sur les règles a pour objectif principal d'identifier toutes les contraintes sur les règles. Ces contraintes sont imposées par le mode d'articulation du bloc qui regroupe les règles. Cependant, cela nécessite également l'identification des contraintes sur les blocs, car ces contraintes peuvent donner lieu à des contraintes sur les règles.

Cette étape se déroule donc comme une cascade de dominos. Elle commence par un parcours des blocs racines qui ne sont pas articulés dans d'autres blocs. Pour chaque bloc racine parcouru, une sous-étape « ajout des contraintes aux blocs et/ou règles dans un bloc B » est appelée.

Cette sous-étape d'« ajout des contraintes aux blocs et/ou règles dans un bloc B » a pour objectif d'identifier les contraintes sur les règles et les blocs dans un bloc B et d'ajouter ces contraintes dans la liste de contraintes Lc. Les règles de ce bloc sont articulées selon un mode m, $m \in \{\text{simultané, successif, progressif, interactif}\}$.

Pour ajouter les contraintes aux blocs et/ou règles dans un bloc B, il faut examiner les sources qui imposent ces contraintes. Comme nous l'avons indiqué, les contraintes peuvent être imposées par le mode d'articulation d'un bloc ou les contraintes sur ce bloc. Il existe donc des contraintes associées au mode d'articulation de B et des contraintes associées au bloc B.

Pour distinguer les contraintes imposées par ces deux sources, nous divisons cette étape en deux sous-étapes : l'ajout des contraintes imposées par le mode d'articulation de B et l'ajout des contraintes imposées par les contraintes sur B.

Les algorithmes de cette étape d'« identification des contraintes sur les règles » sont donnés dans l'annexe B.1 page 208.

4.2.4 Application des contraintes aux règles

L'étape d'application des contraintes aux règles a pour objectif d'appliquer l'ensemble des contraintes aux règles concernées. Ces contraintes effectuent la modification et la génération des règles. Ces contraintes sont imposées par le mode d'articulation du bloc qui regroupe les règles. Cette étape nécessite également l'identification des contraintes sur les blocs car ces contraintes peuvent donner lieu à des contraintes sur les règles.

Cette étape commence par un parcours de toutes les règles de la liste Lr. Pour chaque règle parcourue, une sous-étape « application des contraintes sur la règle R » est appelée.

Cette sous-étape a pour objectif d'appliquer les contraintes sur la règle R. Pour distinguer les deux moyens d'application des contraintes, nous divisons cette étape en deux sous-étapes « génération / modification des éléments aLDEAS » « génération d'un message de notification des contraintes non-respectées ». Ces deux sous-étapes sont effectuées en parallèle et

contenues par un même algorithme qui parcourt toutes les contraintes sur la règle R. La sous-étape de « génération / modification des éléments aLDEAS » permet de générer et de modifier les événements, les conditions, ainsi que les règles. La sous-étape de « génération d'un message de notification » permet d'identifier quelles contraintes sur R ne sont pas respectées. Elle cumule toutes les contraintes non-respectées pour générer un message fourni au concepteur de l'assistance.

Les algorithmes de cette étape sont donnés dans l'annexe B.1 page 208.

4.2.5 Exemple de mise en œuvre du processus d'application des contraintes

Pour montrer comment s'utilisent nos modèles, nous reprenons l'exemple de l'assistance aux utilisateurs de PhotoScape désirant corriger les yeux rouges sur une photo, exemple présenté dans la section 4.1.6.

Une fois l'assistance définie sous forme de trois blocs (cf. Figure 90) par le concepteur de l'assistance grâce notamment au modèle d'articulation entre règles, notre processus d'application des contraintes est exploité pour compléter le système d'assistance et le rendre conforme au langage aLDEAS. Nous déroulons ci-dessous l'exécution de ce processus :

Étape d'initialisation des listes

$L_r = \{R_1, R_2, R_3, R_4, R_5\}$; $L_b = \{B_1, B_2, B_3\}$; $L_c = \{\}$; $L_{bp} = \{\}$

Étape d'identification des contraintes sur les règles

B_3 est un bloc racine.

Étape d'identification des contraintes sur les règles

Ajout des contraintes aux blocs et aux règles dans B_3

$L_{bp} = \{B_3\}$

Comme le mode de B_3 est progressif, les contraintes sur la nouvelle règle d'entrée R_6 du bloc B_3 sont ajoutées à L_c : $C_{creation_regle}(R_6)$, $C_{debut_regle}(R_6, E_3)$, $C_{nb_declenchements}(R_6)$, $C_{alternative}(R_6, B_1, [1,1])$, $C_{alternative}(R_6, B_2, [2,2])$.

Les contraintes sur les règles et blocs articulés sont ajoutées à L_c : $C_{lancement_regle}(B_1)$, $C_{lancement_regle}(B_2)$.

Comme il n'y a pas de contrainte sur B_2 , il n'y a pas d'autre contrainte déduite ajoutée à L_c .

Comme B_3 comporte deux blocs B_1 et B_2 , l'ajout des contraintes aux blocs et aux règles de B_1 et B_2 est effectué.

Ajout des contraintes aux blocs et aux règles dans B_1

$L_{bp} = \{B_3, B_1\}$

Comme le mode de B_1 est successif, les contraintes sur les règles et blocs articulés sont ajoutés à L_c : $C_{fin_regle}(R_1)$, $C_{fin_regle}(R_2)$, $C_{suite_de_regle}(R_2, R_1)$, $C_{suite_de_regle}(R_3, R_2)$.

Comme il y a une contrainte sur B_1 , $C_lancement_regle(B_1)$, une contrainte déduite $C_lancement_regle(R_1)$ est ajoutée à L_c .

Il n'y a aucun bloc dans B_1 .

Ajout des contraintes aux blocs et aux règles dans B_2

$L_{bp}=\{B_3, B_1, B_2\}$

Comme le mode de B_2 est simultané, les contraintes sur la nouvelle règle d'entrée R_7 du bloc B_2 sont ajoutées à L_c : $C_creation_regle(R_7)$, $C_debut_regle(R_7, E_2)$, $C_alternative(R_7, R_4, nul)$, $C_alternative(R_7, R_5, nul)$.

Les contraintes sur les règles et blocs articulés sont ajoutées à L_c : $C_lancement_regle(R_4)$, $C_lancement_regle(R_5)$.

Comme il n'y a pas de contrainte sur B_2 , il n'y a pas d'autre contrainte déduite ajoutée à L_c .

Comme il y a une contrainte sur B_2 , $C_lancement_regle(B_2)$, une contrainte déduite $C_lancement_regle(R_7)$ est ajoutée à L_c .

Il n'y a aucun bloc dans B_2 .

À la fin de l'identification des contraintes, la liste de contraintes L_c contient :

R_1 : $C_fin_regle(R_1)$, $C_lancement_regle(R_1)$

R_2 : $C_fin_regle(R_2)$, $C_suite_de_regle(R_2, R_1)$

R_3 : $C_suite_de_regle(R_3, R_2)$

R_4 : $C_lancement_regle(R_4)$

R_5 : $C_lancement_regle(R_5)$

R_6 : $C_creation_regle(R_6)$, $C_debut_regle(R_6, E_3)$, $C_nb_declenchements(R_6)$, $C_alternative(R_6, B_1, [1,1])$, $C_alternative(R_6, B_2, [2,2])$

R_7 : $C_creation_regle(R_7)$, $C_debut_regle(R_7, E_2)$, $C_alternative(R_7, R_4, nul)$, $C_alternative(R_7, R_5, nul)$

Étape d'application des contraintes sur les règles

L'application des contraintes est effectuée sur chaque règle dans L_c .

Application des contraintes sur R_1

$C_lancement_regle(R_1)$: un événement « lancement par une règle » est ajouté comme événement déclencheur dans R_1 .

$C_fin_regle(R_1)$: comme R_1 ne contient pas d'événement de fin, un message est ajouté dans L_m .

Application des contraintes sur R_2

$C_suite_de_regle(R_2, R_1)$: un événement « fin de R_1 » est créé et ajouté comme événement de fin dans R_2 .

$C_fin_regle(R_2)$: comme R_2 contient un événement de fin, aucun message n'est ajouté dans L_m .

Application des contraintes sur R₃

C_suite_de_regle(R₃, R₂) : un événement « fin de R₂ » est créé et ajouté comme événement de fin dans R₃.

Application des contraintes sur R₄

C_lancement_regle(R₄) : un événement « lancement par une règle » est ajouté comme événement déclencheur dans R₄.

Application des contraintes sur R₅

C_lancement_regle(R₅) : un événement « lancement par une règle » est ajouté comme événement déclencheur dans R₅.

Application des contraintes sur R₆

C_creation_regle(R₆) : la règle R₆ est réellement créée.

C_debut_regle(R₆, E₃) : l'événement E₃ est ajouté comme événement déclencheur dans R₆.

C_nb_declenchements(R₆) : la consultation du nombre de déclenchements de R₆ est créée et ajoutée comme condition dans R₆.

C_alternative(R₆, B₁, [1,1]), C_alternative(R₆, B₂, [2,2]) : deux alternatives pour lancer la règle d'entrée R₁ du bloc B₁ et la règle R₇ du bloc B₂ sont ajoutées dans R₆.

Application des contraintes sur R₇

C_creation_regle(R₇) : la règle R₇ est vraiment créée.

C_debut_regle(R₇, E₂) : l'événement E₂ est ajouté comme événement déclencheur dans R₇

C_alternative(R₇, R₄, nul), C_alternative(R₇, R₅, nul) : une seule alternative pour lancer les deux règles R₄ et R₅ sont ajoutées dans R₇.

4.3. SEPIA-edu : mise en œuvre de l'articulation entre règles

Dans la première version de SEPIA issue de la thèse de Blandine Ginon (Ginon, 2014), l'éditeur d'assistance de SEPIA permettait aux concepteurs d'assistance de créer un système d'assistance en définissant des règles (cf. bas de la Figure 93). Ces règles d'assistance étaient listées dans un tableau de règles (cf. Figure 93). Pour pouvoir comprendre le déroulement d'une assistance complexe, les concepteurs devaient soigneusement examiner le contenu des règles. Pour pouvoir en définir un, ils devaient créer soigneusement l'ensemble des éléments aLDEAS et des règles nécessaires. Ces deux tâches devaient être réalisées entièrement manuellement. Dans le cadre de notre thèse, dans l'objectif de rendre le système plus accessible à des utilisateurs non-informaticiens issus du monde de l'éducation, nous avons complété l'éditeur d'assistance de SEPIA en implémentant les deux modèles que nous venons de présenter : le modèle d'articulation entre règles et le modèle d'application automatique des contraintes des modes d'articulation dans SEPIA. Nous avons également mis en œuvre le processus d'application automatique des contraintes des modes d'articulation présenté dans ce chapitre. Afin de faciliter la distinction entre la version de SEPIA issue de la thèse de Blandine

Ginon et la version de SEPIA modifiée dans le cadre de notre thèse, nous nommerons par la suite SEPIA le système initial et SEPIA-edu le système tel que nous l'avons modifié et enrichi.

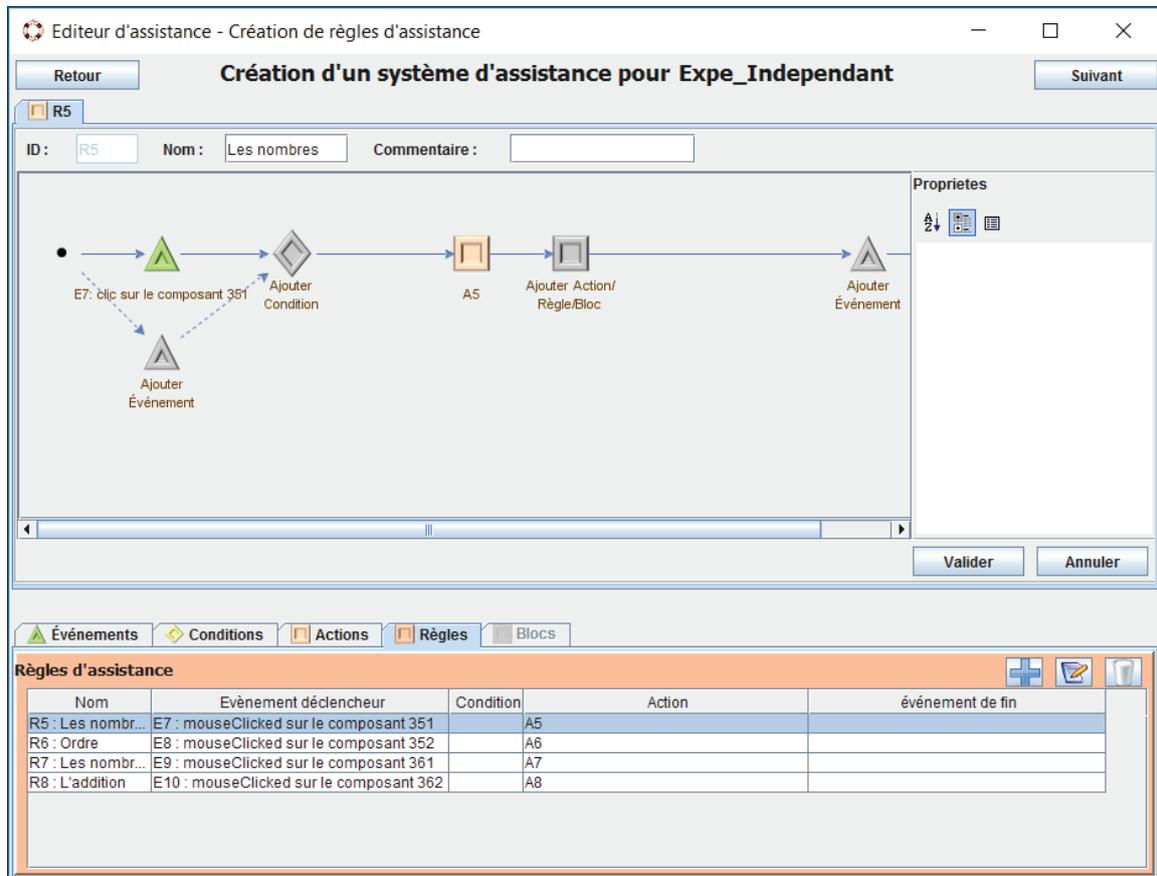


Figure 93 : Définition d'une assistance avec la version initiale du système SEPIA (Ginon, 2014)

Pour expliciter l'articulation entre les règles, nous avons implémenté notre modèle d'articulation entre règles. Cette implémentation a donné lieu à l'ajout d'une fonctionnalité de « définition de blocs de règles articulées en un mode » qui est dédiée au concepteur d'assistance. L'éditeur permet donc désormais aux concepteurs d'assistance de définir explicitement l'articulation entre les règles en définissant des blocs de règles selon cinq modes d'articulation, conformément à notre modèle d'articulation entre règles.

Pour générer et modifier automatiquement les éléments grâce aux blocs définis, nous avons également implémenté notre processus d'application automatique des contraintes aux règles.

L'implémentation de ces deux modèles a eu des conséquences sur l'architecture de SEPIA (cf. Figure 94), que nous avons fait évoluer pour expliciter et faciliter la définition de l'articulation entre règles. Cette évolution ne concerne pas uniquement l'éditeur de SEPIA, elle donne également lieu à deux nouvelles tâches : la « définition de blocs de règles aLDEAS » et l'« application des contraintes sur les règles aLDEAS ».

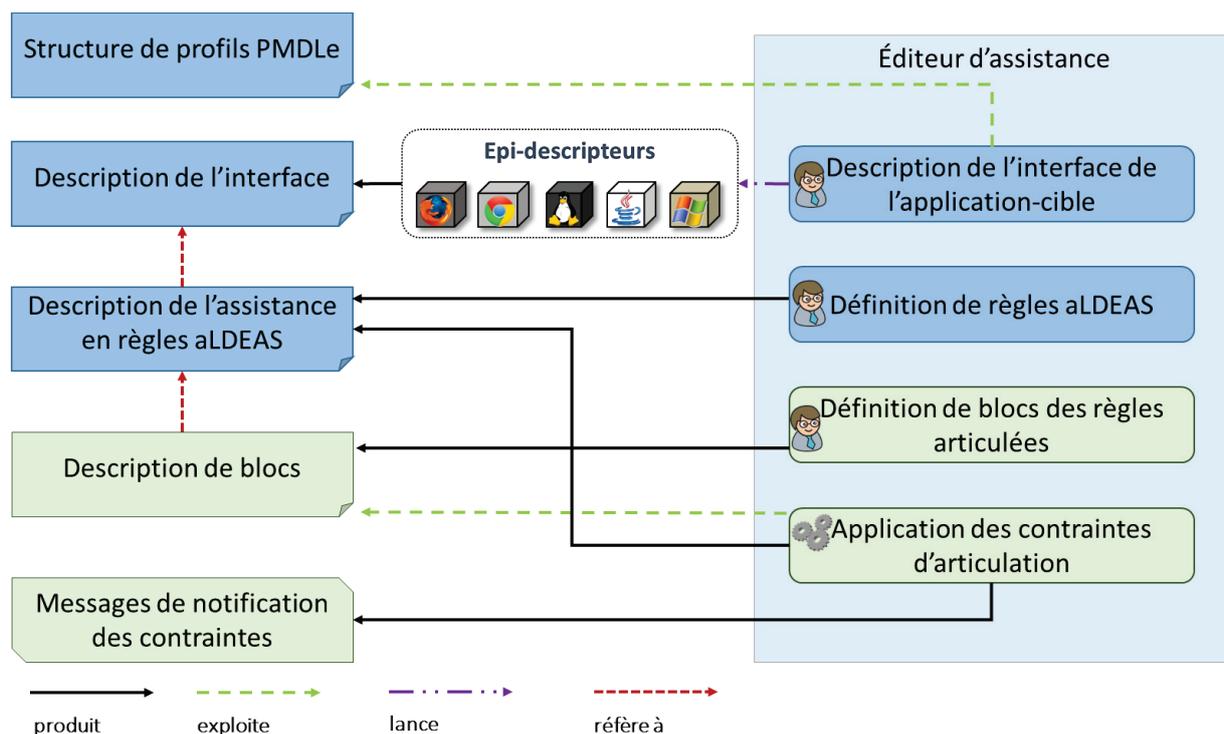


Figure 94 : Architecture de l'éditeur de SEPIA-edu adoptant le modèle d'articulation entre règles et le processus d'application automatique des contraintes

4.3.1 Définition de blocs de règles aLDEAS

L'implémentation du modèle d'articulation entre règles s'est fait en intégrant la définition de blocs de règles dans l'éditeur d'assistance de SEPIA-edu (cf. Figure 94). Cette fonctionnalité est dédiée au concepteur d'assistance. Elle lui permet d'explicitier la définition de l'articulation entre règles grâce à la création de blocs de règles articulées en un mode d'articulation donné. Ainsi, au niveau de l'interface, la notion de bloc de règles remplace pour le concepteur de l'assistance, la notion d'articulation, complexe pour les enseignants, utilisée dans nos modèles. Cette fonctionnalité est nécessaire pour fournir un moyen de visualiser explicitement les blocs de règles et un moyen de manipuler les règles dans un bloc. De plus, elle permet de spécifier les informations liées au bloc pour appliquer automatiquement les contraintes sur les règles.

Cette étape de définition de blocs de règles est réalisée par le concepteur de l'assistance après l'étape de description de l'interface de l'application-cible mais avant, après ou pendant l'étape de définition de règles aLDEAS. Elle est considérée comme une alternative qui remplace une partie de la définition des règles aLDEAS.

La définition de blocs se fait en deux sous-étapes principales : la création d'un nouveau bloc à définir, puis la définition de son contenu. Le résultat produit à l'issue de cette étape est la description des blocs de règles aLDEAS.

4.3.1.1 Création d'un nouveau bloc

Dans cette étape, le concepteur peut créer un nouveau bloc. Pour cela, il doit préciser le mode d'articulation associé à ce bloc. Cinq boutons correspondants aux cinq modes ont été

ajoutés à l'éditeur de SEPIA-edu pour permettre la création d'un nouveau bloc (cf. Figure 95). Ensuite, le concepteur de l'assistance doit préciser le nom, ainsi que la description du bloc si besoin. Pour exploiter un bloc existant, le concepteur choisit un bloc depuis la liste de blocs. Le concepteur pourra ensuite changer le nom, la description ou même le mode d'articulation d'un bloc si besoin.

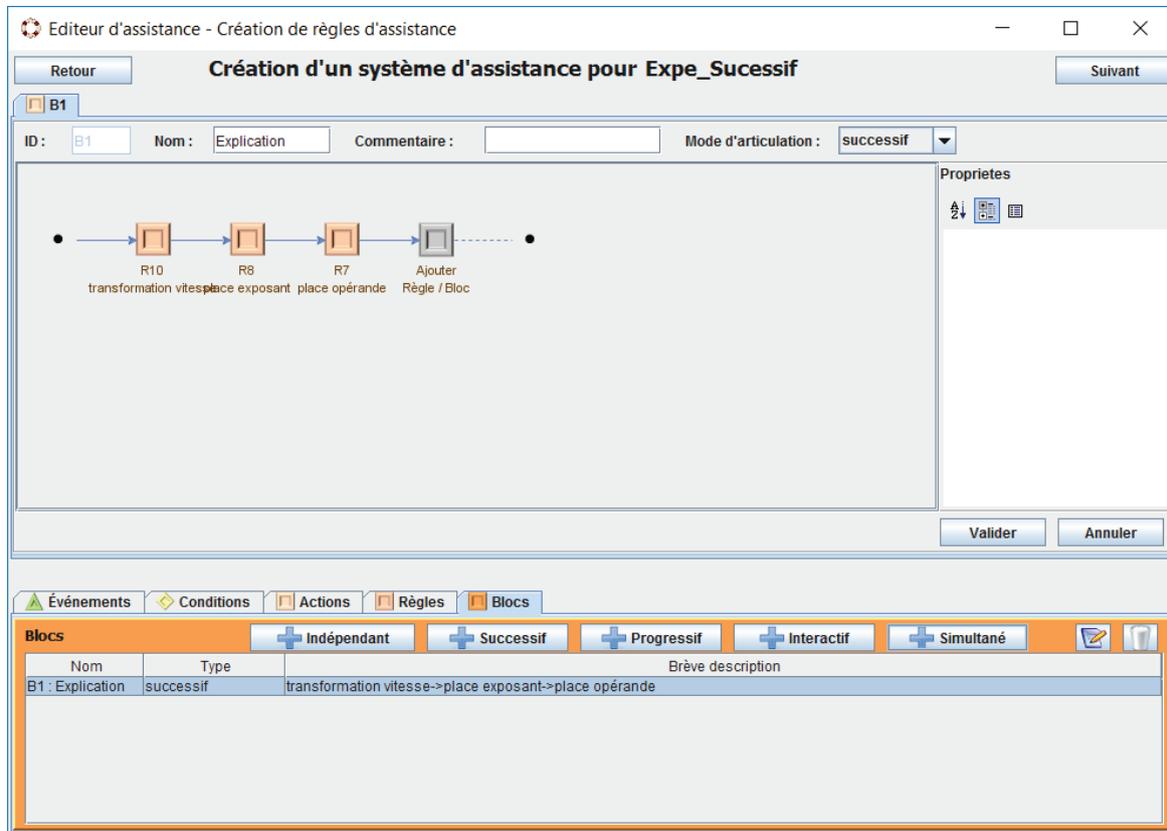


Figure 95 : Définition d'un bloc de règles articulées en mode successif dans SEPIA-edu

4.3.1.2 Définition du contenu d'un bloc

Après la création d'un bloc, le concepteur définit le contenu de celui-ci. Une interface graphique a été implémentée pour permettre cette définition. Cette interface permet une représentation graphique d'un bloc conforme à la représentation proposée dans notre modèle d'articulation entre règles (cf. Figure 80).

Dans cette étape, le concepteur peut manipuler des règles ou des blocs à l'intérieur du bloc qu'il est en train de définir. Il peut s'agir de règles ou de blocs existants, comme de nouvelles règles ou de nouveaux blocs à créer. À travers l'interface graphique, le concepteur peut déplacer facilement les objets graphiques représentant les règles, les blocs et les autres éléments aLDEAS. Il peut préciser d'autres informations dans le tableau de propriétés, par exemple la valeur des alternatives d'une consultation. La partie centrale de la Figure 95 illustre l'interface graphique permettant de définir le contenu d'un bloc de règles articulées en mode successif.

4.3.1.3 Description des blocs

```
<regles>
  <regle id="R7" exigePar="0|B1" nom="Opérande" nomPresentation="place opérande" commentaire="">
    <evenement_de_fin idEve="E4" type="declenchement" idComp="R7" duree="5000" />
    <alternatives condition="">
      <alternative>
        <action id="A8" />
        <action id="A7" />
      </alternative>
    </alternatives>
    <evenement_declencheur idEve="EG1" type="fin" idComp="R8" exigePar="B1" />
  </regle>
  <regle id="R8" exigePar="0|B1" nomPresentation="place exposant" commentaire="">
    <alternatives condition="">
      <alternative>
        <action id="A10" />
        <action id="A9" />
      </alternative>
    </alternatives>
    <evenement_de_fin idEve="E4" type="declenchement" idComp="R8" duree="5000" />
    <evenement_declencheur idEve="EG0" type="fin" idComp="R10" exigePar="B1" />
  </regle>
  <regle id="R9" exigePar="0" nomPresentation="ajout bouton aide" commentaire="">
    <alternatives>
      <alternative>
        <action id="A11" />
      </alternative>
    </alternatives>
    <evenement_declencheur idEve="E0" type="lancementAssistance" idComp="" />
  </regle>
  <regle id="R10" exigePar="0|B1" nomPresentation="transformation vitesse" commentaire="">
    <alternatives condition="">
      <alternative>
        <action id="A12" />
      </alternative>
    </alternatives>
    <evenement_de_fin idEve="E7" type="declenchement" idComp="R10" duree="7000" />
    <evenement_declencheur idEve="E5" type="mouseClicked" idComp="C0" />
  </regle>
</regles>
<blocs>
  <bloc id="B1" dejaGenerere="NON" modeArticulation="successif" nomPresentation="Explication" commentaire="">
    <listeRegles>
      <regle id="R10" indice="0" />
      <regle id="R8" indice="1" />
      <regle id="R7" indice="2" />
    </listeRegles>
  </bloc>
</blocs>
```

Figure 96 : Description d'un bloc de règles stockée dans le fichier de description des règles de SEPIA-edu

La description des blocs est le résultat de la définition des blocs de règles. Elle contient les blocs et leur contenu. Comme un bloc contient des règles et/ou des blocs, cette description réfère à la description des règles.

Pour maintenir la cohérence avec la description des règles déjà existante, compréhensible et exploitable par SEPIA, la description de blocs est formalisée en XML. Pour simplifier le stockage de cette description, nous avons utilisé le même fichier que celui contenant la description des règles. À titre d'exemple, la Figure 96 représente une partie du fichier de description des règles aLDEAS. Il s'agit d'une liste de règles (partie haute de la figure) et d'un bloc (partie basse de la figure) qui fait référence à ces règles articulées en mode successif.

4.3.2 Application des contraintes sur les règles aLDEAS

Pour assurer l'articulation entre les règles, les contraintes d'articulation doivent être appliquées sur les règles aLDEAS. Pour automatiser cette étape, nous avons implémenté notre processus d'application automatique des contraintes d'articulation dans l'éditeur d'assistance de SEPIA-edu. Cette étape permet de minimiser les tâches demandées au concepteur et

d'assurer une articulation correcte des règles contenues dans les différents blocs définis par ce concepteur.

Cette étape est lancée lors de la validation de la création, de la modification ou de la suppression d'un bloc ou d'une règle. Le processus d'application des contraintes a été défini pour permettre une application des contraintes pour tous les blocs et toutes les règles qui définissent un système d'assistance. Cependant, en pratique, dans l'éditeur d'assistance de SEPIA-edu, cette application concerne seulement les blocs et les règles impactés par la création, la modification ou la suppression d'un bloc ou d'une règle étudié.

L'application des contraintes donne lieu à la génération et/ou à la modification automatique des règles aLDEAS concernées et à la notification au concepteur de l'assistance des contraintes non-respectées.

4.3.2.1 Génération et modification des règles aLDEAS

Selon notre modèle d'application des contraintes d'articulation, les éléments aLDEAS pouvant être générés ou modifiés sont les trois principaux éléments d'aLDEAS : règle, consultation et événement, par exemple la génération d'une règle d'entrée pour les modes simultané, progressif et interactif ; la génération de la consultation du nombre de déclenchements d'une règle pour le mode progressif ; la génération d'un événement de fin « fin d'une règle » pour le mode successif ou la modification de règles pour tous les modes.

Afin de ne pas surcharger l'interface proposée au concepteur de l'assistance, les éléments générés ne sont pas ajoutés à l'interface. Ils sont créés uniquement pour être exploités par le moteur d'assistance afin de mettre en œuvre l'articulation entre les règles souhaitée le concepteur d'assistance lors de la définition des blocs de règles. Par exemple, dans la Figure 97, les événements générés EG₁, EG₀ sont présents dans le fichier de description des règles aLDEAS (cf. partie haute de la figure) mais n'apparaissent pas dans l'éditeur d'assistance de SEPIA-edu (cf. partie basse de la figure).

The screenshot shows the XML code for events and a table of generated events. The XML code is as follows:

```

<evenements>
  <evenement type="lancementAssistance" typeEve="lancementAssistance" id="E0" idComp="" exigePar="0|B1" />
  <evenement type="lancementRegle" typeEve="lancementRegle" id="E1" idComp="" exigePar="0|B1" />
  <evenement id="E4" typeEve="timer fixe" duree="5000" unite="secondes" exigePar="0|B1" />
  <evenement typeEve="action utilisateur" id="E5" type="mouseClicked" idComp="C0" exigePar="0|B1" />
  <evenement id="EG1" typeEve="evenement assistance" type="fin" objet="R8" exigePar="B1" />
  <evenement id="E7" typeEve="timer fixe" duree="7000" unite="secondes" exigePar="0|B1" />
  <evenement id="EG0" typeEve="evenement assistance" type="fin" objet="R10" exigePar="B1" />
</evenements>

```

The table below shows the generated events:

Nom	Type	Brève description
E0	événement d'assist...	Lancement de l'assistance
E1	événement d'assist...	Lancement par une autre règle
E4	durée fixe	5 secondes
E5	action de l'utilisateur	clic sur le composant C0
E7	durée fixe	7 secondes

Figure 97 : Aperçu des événements générés par l'éditeur d'assistance de SEPIA-edu pour mettre en œuvre l'articulation entre les blocs d'un système d'assistance

4.3.2.2 Notification des contraintes non-respectées

L'application des contraintes d'articulation ne peut pas être complètement automatisée. Cette limite concerne les contraintes imposant de posséder au moins un événement de fin sur une règle. La liste des contraintes non-respectées est donc fournie au concepteur d'assistance au travers d'un message afin qu'il puisse corriger les éléments aLDEAS correspondants. Par exemple, la Figure 98 montre le message de notification d'une contrainte non-respectée pour la règle R₈.

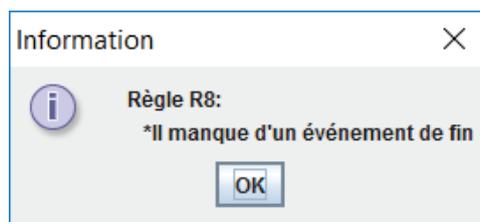


Figure 98 : Message de notification de contrainte non-respectée dans SEPIA-edu

4.4. Exemple d'un déroulement d'une assistance

Dans cette section, nous présentons un exemple du déroulement d'une assistance avec l'éditeur SEPIA-edu. Ce système d'assistance est donc constitué de blocs de règles articulées.

Notre exemple est celui d'un système d'assistance pour MatouMatheux - un site Web proposant des exercices en mathématiques. Ce système d'assistance aide les apprenants à réaliser un exercice sur la proportionnalité en fournissant progressivement une explication et un exemple. L'exécution de ce système est illustrée par une vidéo en ligne disponible sur le site du projet AGATE¹ (cf. Figure 99).

Nous allons à présent détailler le contenu de ce système d'assistance, son déroulement et montrer pour chaque étape comment elle est mise en œuvre à l'aide des blocs et règles définis selon le langage aLDEAS.

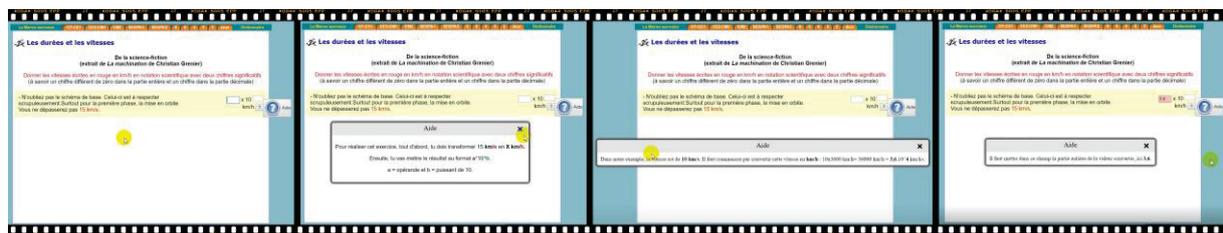


Figure 99 : Vue d'ensemble du système d'assistance qui ajoute un bouton d'aide et propose progressivement une explication et un exemple lors des clics sur ce bouton d'aide

- Étape 1 (cf. Figure 101) : Lors du lancement de l'assistance, le système d'assistance ajoute un bouton d'aide à côté du bouton « ? » dans l'interface du site MatouMatheux.

¹ Lien de la vidéo de l'exécution du système d'assistance de MatouMatheux : <https://www.youtube.com/watch?v=BXKjEKyrlVA>

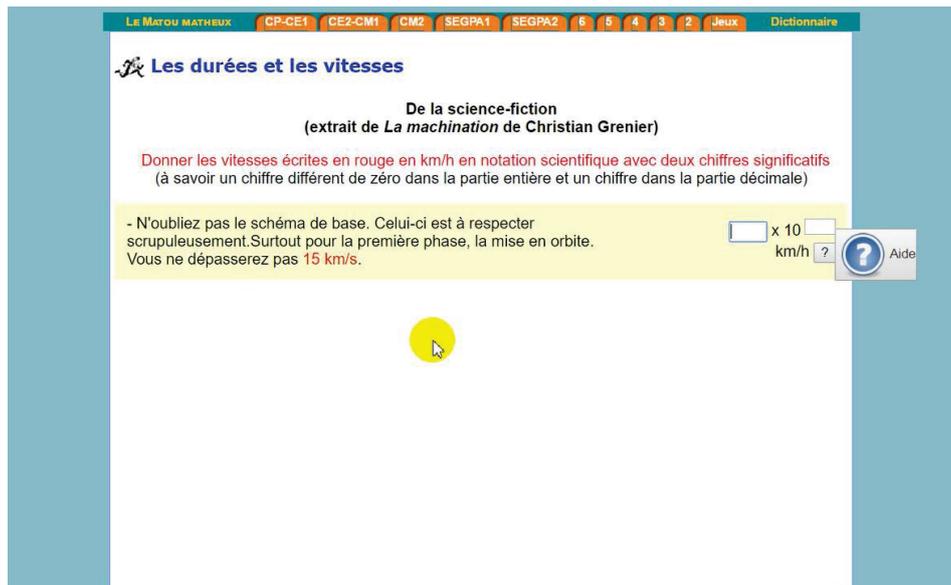


Figure 100 : Exécution de l'étape "ajout d'un bouton d'aide" du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu

En aLDEAS :

R₉ (cf. Figure 101) : la règle R₉ débute au lancement de l'assistance. R₉ effectue l'action A₈ pour ajouter un bouton d'aide.

A₈ : type d'action = « ajout d'un composant », composant concerné = « 3 » (bouton « ? »), direction = « par la droite », identifiant attribué au composant ajouté = « C₀ »

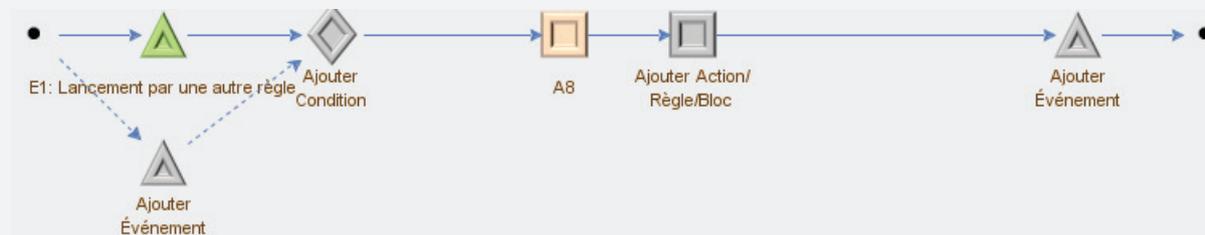


Figure 101 : Définition de la règle R₉ avec l'éditeur SEPIA-edu pour le système d'assistance de MatouMatheux

- Étape 2 : Lors des clics sur le bouton d'aide, le système d'assistance fournit à l'utilisateur une explication (cf. Figure 104) et un exemple (cf. Figure 107, Figure 109, Figure 111) de façon progressive.

En aLDEAS :

Bloc de règles : B₁ (cf. Figure 102)

Mode d'articulation : progressif

Règles/blocs articulés : R₉ (étape 2.1 : donner une explication), B₂ (étape 2.2 : donner un exemple)

Règles générées : RG₀ (cf. Figure 103)

Règles modifiées : R₉ (cf. Figure 105)

Conditions générées : CG₀ (consultation du nombre de lancements de RG₀).

Le bloc de règles B₁ débute par l'événement E₄ (lors du clic sur le bouton d'aide). B₁ permet de lancer les règles progressivement. B₁ consulte donc le nombre de lancements de ce bloc. En fonction du nombre de lancements, B₁ lance la règle R₉ ou le bloc de règles B₂ pour fournir respectivement une explication ou un exemple.

Application des contraintes : lors de la validation du bloc B₁

- La consultation CG₀ est automatiquement générée
- La règle d'entrée RG₀ est automatiquement générée

E₄ : type d'événement = « action de l'utilisateur », action = « clic », composant = « C₀ » (bouton d'aide)

CG₀ : type de consultation = « consultation de l'historique de l'assistance », sous-type = « nombre de déclenchements », élément aLDEAS = « EG₀ »

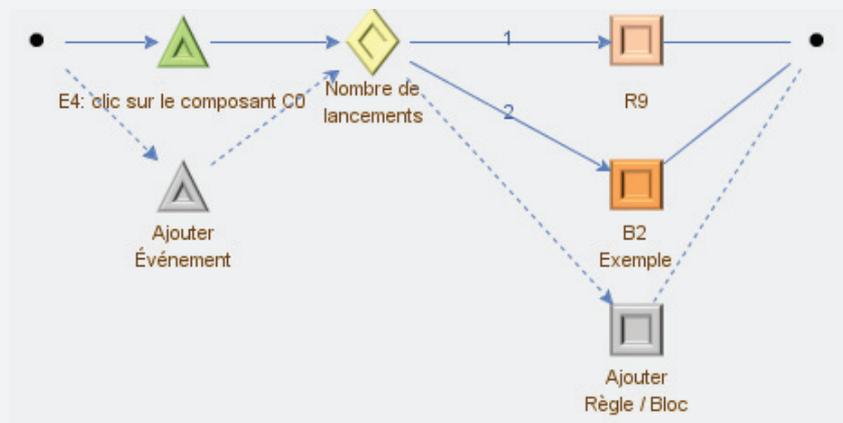


Figure 102 : Définition du bloc de règles B₁ pour MatouMatheux avec l'éditeur SEPIA-edu

```
<regle id="RG0" exigePar="B1" regleDebutDe="B1">
  <evenement_declencheur idEve="E4" type="mouseClicked" idComp="C0" exigePar="B1" />
  <alternatives condition="CG0">
    <alternative borne_inf="1" borne_sup="1">
      <action id="R9" />
    </alternative>
    <alternative borne_inf="2" borne_sup="2">
      <action id="R10" />
    </alternative>
  </alternatives>
</regle>
```

Figure 103 : Génération de la règle RG₀ (invisible pour les concepteurs d'assistance) avec l'éditeur SEPIA-edu pour le système d'assistance de MatouMatheux

- Étape 2.1 : Lors du premier clic sur le bouton d'aide, le système affiche une explication sur l'exercice.

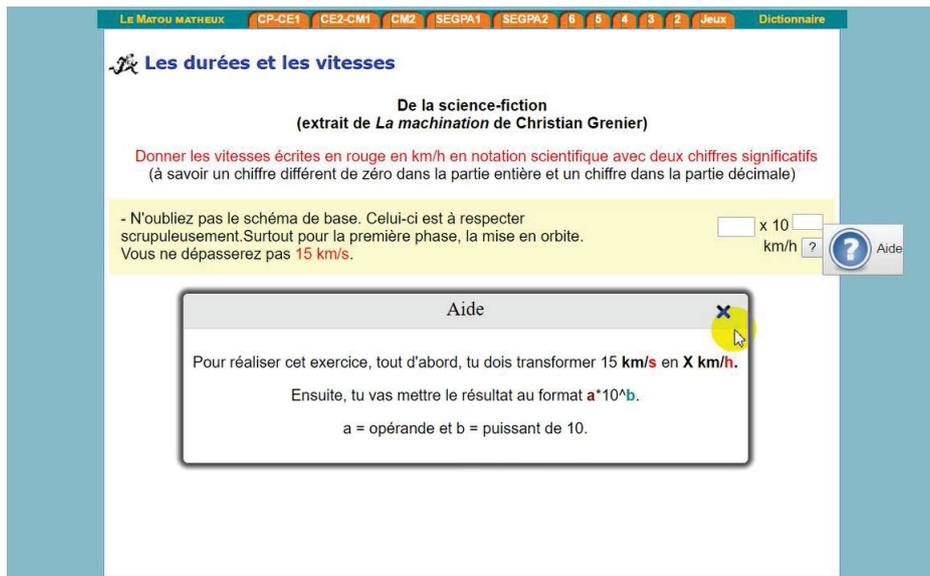


Figure 104 : Exécution de l'étape "affichage d'une explication" du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu

En aLDEAS :

R₉ (cf. Figure 105) : la règle R₉ débute par un événement déclencheur « lancement par une règle ». R₉ effectue l'action A₈ pour afficher une explication sur l'exercice.

Application des contraintes : lors de la validation du bloc B₁ :

- L'événement déclencheur « lancement par une règle » est ajouté à la règle R₉ si elle ne le contient pas encore

A₈ : type d'action = « message », texte = « Pour réaliser cet exercice, tout d'abord, tu dois transformer 15 km/s en X km/h. Ensuite, tu vas mettre le résultat au format a*10^b. a = opérande et b = puissant de 10. »

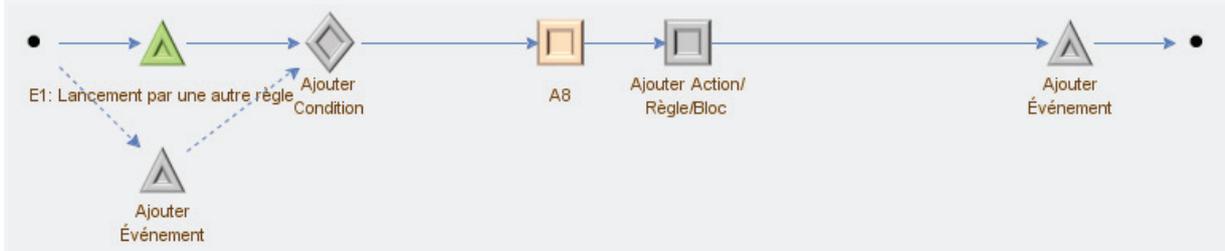


Figure 105 : Définition de la règle R₉ avec l'éditeur SEPIA-edu pour le système d'assistance MatouMatheux

- Étape 2.2 : Lors du deuxième clic, le système affiche un exemple comportant trois étapes successives : exposer l'exemple (cf. Figure 107), remplir le champ « opérande » (cf. Figure 109), remplir le champ « puissance » (cf. Figure 111).

En aLDEAS :

Bloc de règles : B₂ (cf. Figure 106)

Mode d'articulation : successif

Règles/blocs articulés : R₁₀ (étape 2.2.1 : exposer exemple), R₁₁ (étape 2.2.3 : remplir opérande), R₁₂ (étape 2.2.3 : remplir puissance).

Règles modifiées : R₁₀, R₁₁, R₁₂

Le bloc de règles B₂ débute par la règle R₁₀. R₁₀ permet d'afficher l'exemple avec trois étapes successives en lançant trois règles successives : R₁₀, R₁₁, R₁₂. La validation du bloc B₁ permet de modifier ces trois règles.

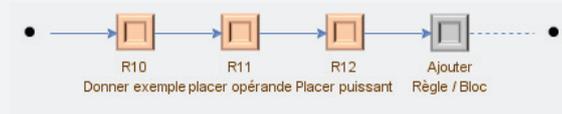


Figure 106 : Définition du bloc de règles B₂ avec l'éditeur SEPIA-edu pour le système d'assistance MatouMatheux

- Étape 2.2.1 (cf. Figure 107) : Pour la première étape de l'exemple, le système fournit un message textuel pour exposer l'exemple. Ce message disparaît après 8 secondes.

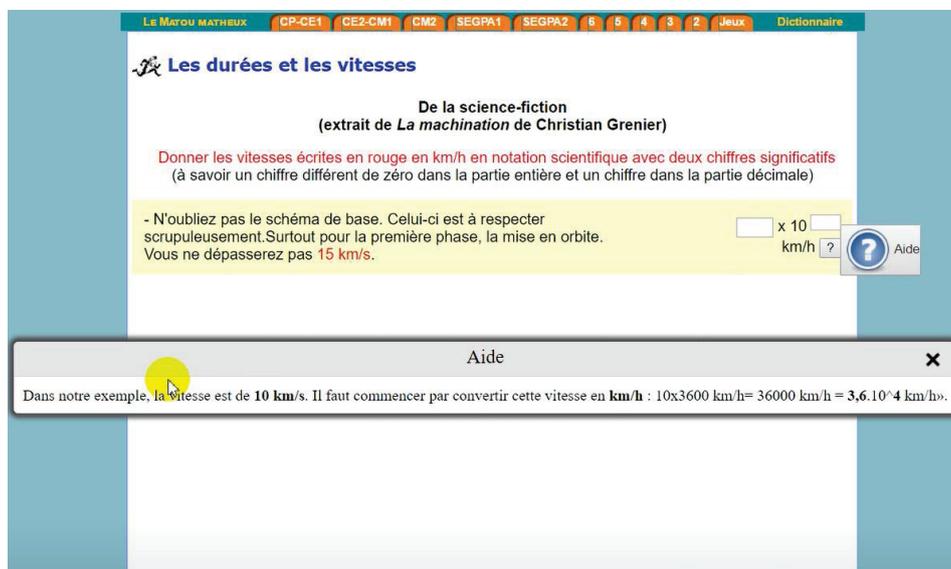


Figure 107 : Exécution de l'étape "affichage d'un exemple" du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu

En aLDEAS:

Lors du deuxième clic sur le bouton d'aide qui est détecté *via* le bloc de règles B₁, l'exemple est fourni *via* le bloc de règles B₂. La première étape de l'exemple est effectuée en premier *via* la règle R₁₀ (cf. Figure 108). Le bloc de règles B₁ (dont la règle RG₀ est une règle d'entrée) lance donc la règle R₁₀ lors du deuxième clic sur le bouton d'aide. R₁₀ débute par un événement déclencheur « lancement par une règle ». R₁₀ effectue l'action A₁₄ pour exposer l'exemple. La règle R₁₀ se termine par l'événement de fin E₈.

Application des contraintes : lors de la validation du bloc de règles B₂, ou lors de la création de la règle R₁₀ depuis le bloc de règles B₂ :

- l'événement « lancement par une règle » est ajouté à la règle R₁₀ si elle ne le contient pas encore.

- un message indiquant l'absence d'un événement de fin est affiché si la règle R₁₁ ne contient pas encore d'événement de fin.

A₁₀ : type d'action = « message », texte = « Dans notre exemple, la vitesse est de 10 km/s. Il faut commencer par convertir cette vitesse en km/h : $10 \times 3600 \text{ km/h} = 36000 \text{ km/h} = 3,6 \cdot 10^4 \text{ km/h}$ »

E₈ : type d'événement = « durée », durée = « 8 secondes », depuis = « lancement de la règle en cours »

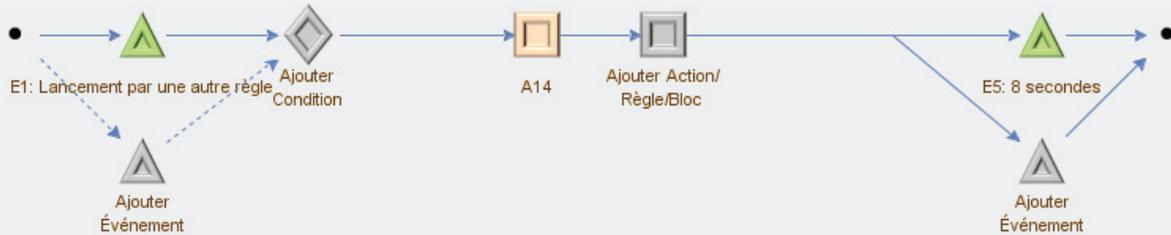


Figure 108 : Définition de la règle R₁₀ avec l'éditeur SEPIA-edu pour le système d'assistance MatouMatheux

➤ Étape 2.2.2 (cf. Figure 109) : Pour la deuxième étape de l'exemple, le système affiche un message, met en valeur le champ « opérande » et remplit automatiquement ce champ. Le message et la mise en valeur sont terminés après 8 secondes.

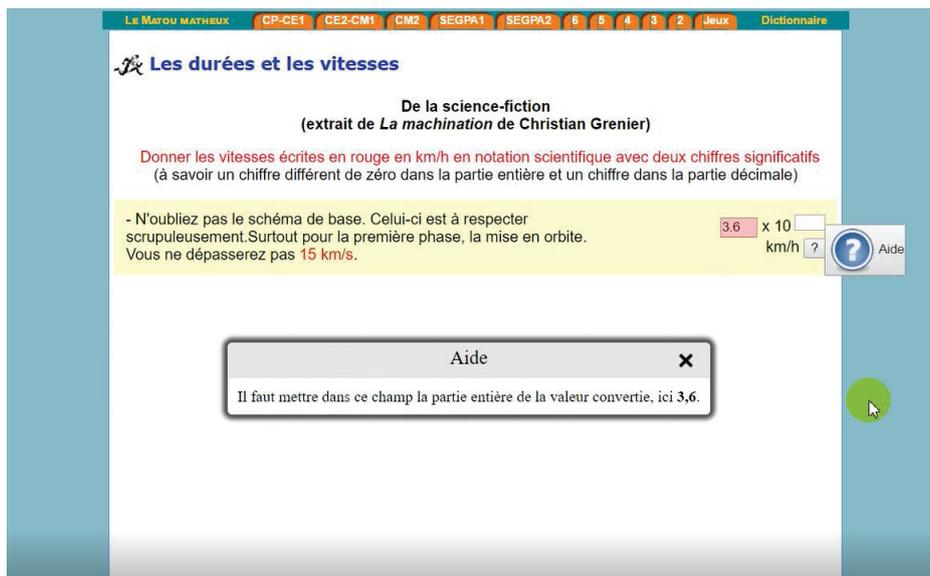


Figure 109 : Exécution de l'étape « remplissage automatique de l'opérande » du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu

En aLDEAS :

Cette étape est effectuée lors de la fin de l'étape 2.2.1 (fin de la règle R₁₀). La règle R₁₁ (cf. Figure 110) débute donc par l'événement déclencheur EG₀ « fin de la règle R₁₀ ». La règle R₁₁ effectue les actions A₁₅, A₁₇ et A₁₈. La règle R₁₁ se termine par l'événement de fin E₈.

Application des contraintes : lors de la validation du bloc de règles B₂, ou lors de la création de la règle R₁₀ depuis le bloc de règles B₂ :

- l'événement déclencheur EG_0 est ajouté à la règle R_{11} si elle ne le contient pas encore
- un message indiquant l'absence d'un événement de fin est affiché si la règle R_{11} ne contient pas encore d'événement de fin.

EG_0 : type d'événement = « lié à l'assistance », sous-type = « fin », élément aLDEAS = « R_{10} »

A_{15} : type d'action = « message », texte = « Il faut mettre dans ce champ la partie entière de la valeur convertie, ici 3,6. ».

A_{17} : type d'action = « mise en valeur », type de mise en valeur = « colorer », composant = « 1 » (champ « opérande »)

A_{18} : type d'action = « action sur l'interface », action = « modifier la valeur », composant = « 1 » (champ « opérande »), valeur attribuée = « 3,6 »

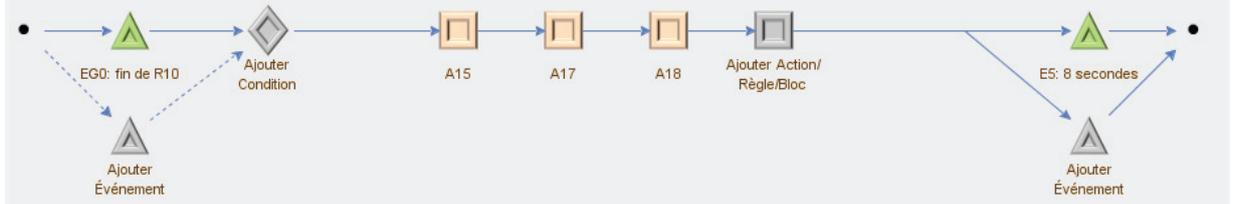


Figure 110 : Définition de la règle R_{11} avec l'éditeur SEPIA-edu pour le système d'assistance MatouMatheux

- Étape 2.2.3 (cf. Figure 111) : Pour la troisième étape de l'assistance de notre exemple, le système affiche un message, met en valeur le champ « puissance » et remplit automatiquement ce champ. Le message et la mise en valeur sont stoppés après 8 secondes.

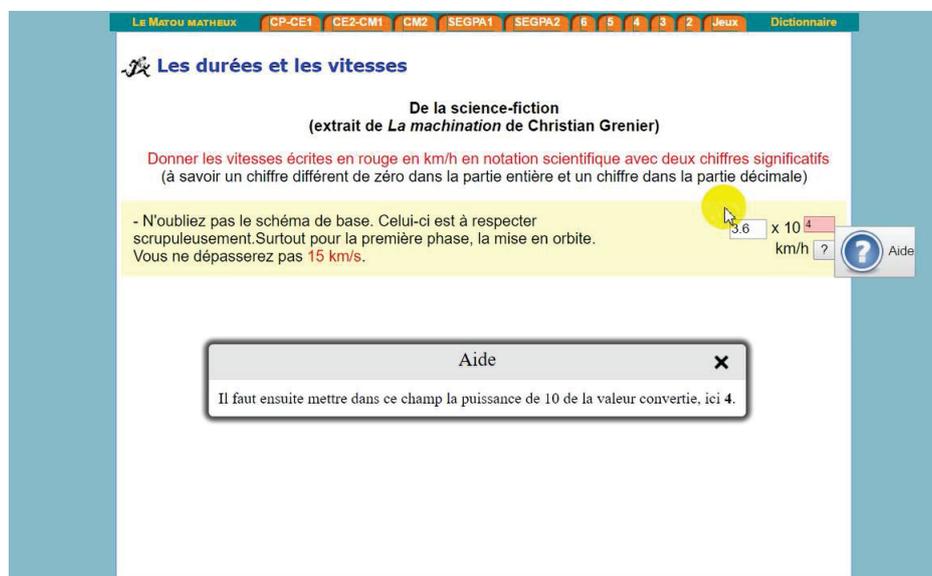


Figure 111 : Exécution de l'étape « remplissage automatique de la puissance » du système d'assistance de MatouMatheux avec la version Web du moteur SEPIA-edu

En aLDEAS :

Cette étape est effectuée lors de la fin de l'étape 2.2.2 (fin de la règle R₁₁). La règle R₁₂ (cf. Figure 112) débute donc par un événement déclencheur EG₁ « fin de la règle R₁₁ ». La règle R₁₂ effectue les actions A₁₆, A₁₉ et A₂₀. La règle R₁₂ se termine par l'événement de fin E₈.

Application des contraintes : lors de la validation du bloc de règles B₂, ou lors de la création de la règle R₁₂ depuis le bloc de règles B₂ :

- l'événement déclencheur EG₀ est ajouté à la règle R₁₂ si elle ne le contient pas encore
- la règle R₁₂ n'est pas contrainte à posséder un événement de fin car elle est lancée en dernier.

EG₀ : type d'événement = « lié à l'assistance », sous-type = « fin », élément aLDEAS = « R₁₁ »

A₁₆ : type d'action = « message », texte = « Il faut ensuite mettre dans ce champ la puissance de 10 de la valeur convertie, ici 4. »

A₁₉ : type d'action = « mise en valeur », type de mise en valeur = « colorer », composant = « 2 » (champ « puissance »)

A₂₀ : type d'action = « action sur l'interface », action = « modifier la valeur », composant = « 2 » (champ « puissance »), valeur attribuée = « 3,6 »



Figure 112 : Définition de la règle R₁₂ avec l'éditeur SEPIA-edu pour le système d'assistance de MatouMatheux

Dans ce chapitre, nous avons proposé d'une part le **modèle d'articulation entre règles aLDEAS** qui permet de définir des blocs de règles articulées selon un mode donné (indépendant, successif, simultané, progressif ou interactif) et d'autre part le **processus d'application des contraintes des modes d'articulation entre règles** qui permet de générer et/ou modifier les règles aLDEAS correspondantes. Ce modèle et ce processus ont été implémentés dans SEPIA-edu sous la forme d'une part d'une étape destinée aux enseignants, de « définition de blocs de règles articulées » et d'autre part d'une étape automatisée d'« application des contraintes ». Au travers d'une interface graphique, les enseignants peuvent donc créer un bloc de règles articulées dans le mode souhaité, qui représente le mode de déroulement d'une assistance.

Chapitre 5. Proposition de Guidage pédagogique

Objectif de ce chapitre

Proposer de nouveaux modèles et outils permettant d'explicitier et de faciliter la définition d'un guidage pédagogique en aLDEAS avec SEPIA-edu.

Introduction de ce chapitre

Dans le Chapitre 3, la confrontation du système SEPIA et du langage aLDEAS aux caractéristiques de l'assistance en contexte éducatif nous a amené à deux questions de recherche : « Comment définir le déroulement d'une assistance avec SEPIA ? » et « Comment définir un guidage pédagogique avec SEPIA ? ». Dans ce chapitre, nous répondons à la deuxième question de recherche.

Nous présentons donc nos propositions qui visent à faciliter la définition d'un guidage pédagogique avec SEPIA-edu. Nous proposons un **modèle d'activité**, un **modèle de guidage pédagogique**, un **patron de guidage pédagogique**, un **processus de transformation d'un guidage pédagogique en règles aLDEAS** compatibles avec le système SEPIA. Ces modèles, ce patron, ce processus ont été **implémentés** dans **SEPIA-edu**. L'**exemple** détaillé présenté à la fin du chapitre montre comment définir un guidage pédagogique avec SEPIA-edu.

Contributions de ce chapitre

Modèle d'activité, modèle de guidage pédagogique, patron de guidage pédagogique, processus de transformation d'un guidage pédagogique en règles aLDEAS et leur mise en œuvre dans SEPIA

Publications liées à ce chapitre :

Thai, L.-V. : Concevoir un guidage pédagogique à travers un système d'assistance épiphyte. Démonstration à RJC-EIAH (2016).

Le guidage pédagogique est l'un des besoins d'assistance spécifiques au contexte éducatif que nous avons identifiés dans notre état de l'art. Cependant, SEPIA ne soutient pas suffisamment la définition du guidage pédagogique pour aider les concepteurs d'assistance non-informaticiens, des enseignants dans le contexte éducatif. Dans la version initiale de SEPIA, la définition d'un guidage pédagogique, tâche complexe, se faisait de manière implicite.

Dans ce chapitre, nous présentons notre seconde contribution au projet AGATE qui correspond à la question de recherche : « comment définir un guidage pédagogique dans SEPIA ? ». Cette contribution théorique se représente sous forme d'un ensemble de modèles, d'un patron et d'un processus. Le premier modèle est le modèle d'activité qui explique comment identifier dans SEPIA-edu une activité contenue dans les applications-cibles. Le second modèle est celui du guidage pédagogique qui explique comment définir cinq types de guidages pédagogiques. Ensuite, le patron de guidage pédagogique facilite la définition d'un guidage pédagogique selon un ou plusieurs des cinq types. Pour finir, le processus de transformation du guidage pédagogique en règles aLDEAS montre comment transformer automatiquement un guidage pédagogique en règles aLDEAS. Le patron et le processus sont implémentés au sein de l'éditeur de SEPIA-edu pour permettre aux enseignants de travailler au maximum sur les objets pédagogiques (activité et guidage) au lieu de manipuler des règles aLDEAS.

5.1. Modèle d'activité

Le guidage pédagogique est la préconisation d'un ensemble organisé d'activités d'apprentissage à effectuer sur une ou plusieurs applications informatiques. Les activités peuvent être préexistantes ou non sur les applications-cibles. Pour pouvoir délimiter ces activités au sein des applications-cibles, les organiser dans un guidage et exécuter ce guidage, nous proposons un modèle d'activité.

Ce modèle d'activité a pour objectif d'identifier l'activité sur les applications-cibles. Il ne permet pas d'explicitement toutes les informations concernant une activité comme le font par exemple les modèles EML (Koper, 2001) et IMS-LD (IMS-LD, 2016), mais uniquement les informations nécessaires pour définir et exécuter un guidage pédagogique dans SEPIA-edu.

Dans ce modèle, une activité est décrite par son **nom** et sa **description**. Alors que son nom permet de distinguer une activité d'une autre, sa description permet de l'explicitement. Ces deux propriétés sont définies textuellement. Cependant, ces deux propriétés ne suffisent pas pour définir son exécution dans le guidage pédagogique avec SEPIA-edu. Deux autres éléments sont donc ajoutés à ce modèle pour permettre son exécution : les **états de sortie** et les **règles associées**.

5.1.1 États de sortie

Un ensemble d'états de sortie permet de définir comment une activité se termine. Une activité peut donc comporter plusieurs états de sortie (par exemple la note, le taux de réussite, le taux de la maîtrise). Ces états sont des éléments qui vont ensuite permettre de contraindre le choix des activités à proposer à l'apprenant afin qu'il soit plus pertinent. Ces états de sortie contribuent donc à la spécification et l'exécution d'un guidage pédagogique personnalisé.

Les états de sortie peuvent être des valeurs textuelles (réussite/échec, maîtrisé/non maîtrisé...), des valeurs numériques (notes sur 10, sur 20...) ou des dates (réalisation de l'activité le 10/10/2016). Ces valeurs sont des éléments qui composent le profil de l'apprenant.

Pour définir les états de sortie de l'activité, nous adoptons le langage PMDLe qui est déjà utilisé pour définir le profil de l'utilisateur dans le but d'adapter l'assistance dans SEPIA. Ce langage de modélisation des profils d'apprenants a été proposé par l'équipe LIRIS-TWEAK dans des travaux de recherche antérieurs (Eyssautier-Bavay, 2008; Jean-Daubias, Ginon and Lefevre, 2011).

5.1.1.1 Le langage PMDLe

Le langage PMDL (Eyssautier-Bavay, 2008; Jean-Daubias, Ginon and Lefevre, 2011) est un langage de modélisation de profils permettant la définition de profils très variés selon un même formalisme. Pour cela, PMDLe est constitué de deux parties différentes : la structure du profil et les données contenues dans un profil. La partie « structure » décrit l'organisation des données du profil. Elle peut être commune à plusieurs apprenants. Cette partie permet de construire correctement la partie « donnée » qui contient les valeurs propres à chaque apprenant.

La structure de profils PMDLe est constituée d'un ensemble d'éléments hiérarchisés nommés briques. Pour définir un ensemble d'états de sortie d'une activité, nous utiliserons le type de briques « liste de composantes ». Ce type permet de structurer les informations sous forme de listes de composantes, hiérarchiques ou non, auxquelles sont associées une ou plusieurs valeurs. Une liste hiérarchique de composantes est une liste dans laquelle les composantes peuvent contenir des sous-composantes qui elles-mêmes peuvent être détaillées en sous-composantes. Les composantes ou sous-composantes sont des terminales si elles ne contiennent pas de sous-composantes. Par exemple, le Tableau 1 montre trois éléments dans les profils d'IHM d'Hélène et de Lucas : « Conception », « Évaluation ergonomique » et « Développement d'application ». La brique « Développement d'application » est constituée de deux composantes : « Application bureau » et « Application Web ». La composante « Application bureau » est détaillée en deux sous-composantes « Application Windows » et « Application Java ».

Afin de définir les valeurs possibles pour les composantes ou sous-composantes, chaque brique est associée à une échelle PMDLe. Une échelle peut être définie par un intervalle de valeurs et un pas. Dans notre exemple, l'échelle pour les notes a un intervalle numérique [0,20] avec un pas de 1. Une échelle peut également être définie par une liste de valeurs. Par exemple, une échelle du niveau de maîtrise est une liste des valeurs textuelles « Non maîtrisé / Partiellement maîtrisé / Maîtrisé ».

Les données du profil sont construites pour chaque apprenant en se basant sur cette structure. Dans le profil d'un apprenant, les composantes et sous-composantes terminales peuvent contenir une ou plusieurs valeurs respectant l'échelle de l'élément du profil. Par exemple, la première brique, « Conception » (cf. Tableau 1) est associée à deux échelles (l'une destinée au niveau de maîtrise, l'autre à la note sur 20), alors que la deuxième brique, « Évaluation ergonomique », est associée à une échelle de niveau de maîtrise et que la dernière brique, « Développement d'applications », est associée à une échelle de notes sur 20. Toutes les composantes terminales « Méthodes de conception », « Technique de recueil d'information », « Bastien et Scapin », « Évaluation Web » et « Application Web », ainsi que les

sous-composantes terminales « Application Windows » et « Application Java », portent une valeur respectant l'échelle associée à leur brique.

	Hélène	Lucas
Conception échelles : (Non maîtrisé / Partiellement maîtrisé / Maîtrisé) ; ([0,20], 1)		
Méthodes de conception	Non maîtrisé	Partiellement maîtrisé
Technique de recueil d'informations	6	1
Évaluation ergonomique échelle : (Non maîtrisé / Partiellement maîtrisé / Maîtrisé)		
Bastien et Scapin	Partiellement maîtrisé	Maîtrisé
Évaluation Web	Partiellement maîtrisé	Maîtrisé
Développement d'applications échelles : ([0,20], 1)		
Application bureau		
Application Windows	11	16
Application Java	10	14
Application Web	9	14

Tableau 1 : Exemples de profils d'utilisateurs respectant le formalisme PMDLe

5.1.1.2 État de sortie en PMLDe

PMDLe étant déjà exploité dans SEPIA pour gérer les profils d'apprenant, et étant adapté à nos besoins concernant les états de sortie, nous avons décidé de modéliser ceux-ci avec PMDLe.

La Figure 113 illustre la définition des états de sortie pour SEPIA-edu en PMDLe. Ainsi, une brique PMLDe représente une activité. Chaque état de sortie porte une valeur (texte, numérique ou date). C'est une composante terminale PMDLe qui représente cet état de sortie. Les valeurs d'un état de sortie doivent respecter l'échelle de la brique.

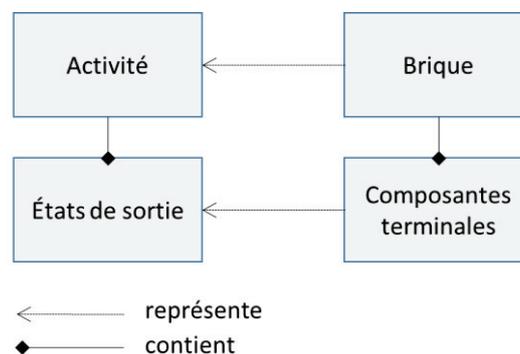


Figure 113 : Définition des états de sortie pour SEPIA-edu en PMDLe

Le Tableau 2 montre un exemple des états de sortie de deux activités A_1 et A_2 correspondant à des exercices en mathématiques. Alors que l'activité A_1 est évaluée en termes de réussite et de notes, l'activité A_2 est évaluée en termes de maîtrise et de temps de réalisation. Chaque état

de sortie « Réussite de A_1 », « Note de l'activité A_1 », etc. porte une valeur qui respecte l'échelle de l'activité.

	Hélène	Lucas
Activité A_1 échelle : (Réussite/Échec) ; ([0,20], 1)		
Note	14	6
Réussite	Réussite	Échec
Activité A_2 échelle : (Non maîtrisé/ Partiellement maîtrisé/ Maîtrisé) ; ([0,30], 1)		
Temps de réalisation en minutes	20	15
Maîtrise	Maîtrisé	Partiellement maîtrisé

Tableau 2 : Exemple de description des états de sortie pour deux activités respectant le formalisme PMDL

5.1.2 Règles associées à une activité

Afin d'organiser les activités, il est nécessaire qu'un système d'assistance sache si les activités se terminent et avec quels états de sortie elles se terminent. Ainsi, l'indication de fin d'activité, associée aux états de sortie de celle-ci, permet au système d'assistance de continuer à proposer aux apprenants d'autres activités.

Par ailleurs, un enseignant pourrait vouloir guider l'apprenant durant la réalisation de l'exercice, il peut donc définir de l'assistance (constituée de règles aLDEAS) pour une activité donnée.

On a donc au final pour une activité, deux catégories de règles associées : les règles, obligatoires, qui mettent fin à l'activité avec la complétion des états de sortie et éventuellement les règles optionnelles qui assistent l'apprenant dans l'activité. Nous allons à présent définir ces deux catégories de règles.

Pour faciliter la définition de ces règles, ainsi qu'augmenter la performance du moteur de SEPIA-edu, nous proposons un nouveau mécanisme pour la phase d'exécution d'un système d'assistance : l'activation / désactivation des règles associées à l'activité. Ce mécanisme assure que les règles qui ne concernent pas à l'activité en cours ne sont pas prises en compte dans la phase d'exécution par le moteur de SEPIA-edu. Ces règles ne sont donc pas lancées, même si leurs événements déclencheurs sont détectés.

5.1.2.1 Règles mettant fin à l'activité avec des états sortie

Les règles permettant de mettre fin à l'activité avec des états sortie sont nécessaires pour le déroulement du guidage pédagogique. Elles permettent de mettre à jour les états de sortie lors de la fin de l'activité.

Ces règles nécessitent l'ajout dans le langage aLDEAS d'une nouvelle action d'assistance « **fin de l'activité** » dans laquelle les états de sortie sont mis à jour. Nous avons ajouté cette action et l'avons positionnée dans la typologie des actions élémentaires d'assistance aLDEAS (cf. Annexe C page 212).

Prenons un exemple : dans la Figure 114, la règle R_1 associée à l'activité A_1 , met fin à cette activité quand l'apprenant clique sur le bouton « validation ». Si la valeur du composant 1 est égale à 1 (l'apprenant a donné une bonne réponse), l'activité prend fin avec succès. Sinon, elle prend fin avec échec. Cette indication de succès ou d'échec sera utilisée dans la suite du déroulement du guidage pédagogique.

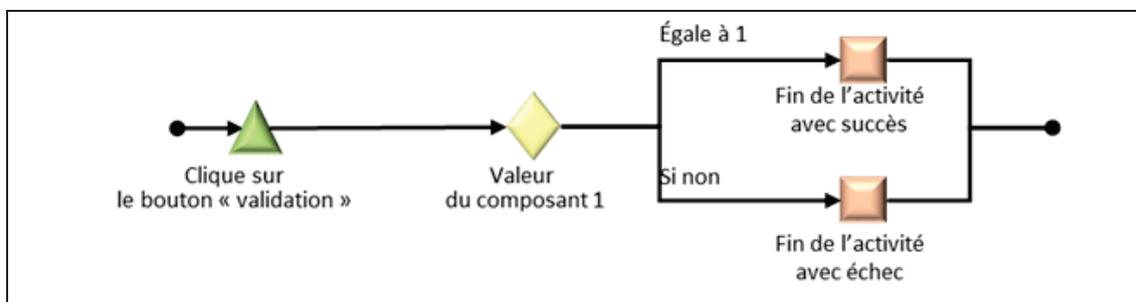


Figure 114 : Exemple d'une règle aLDEAS permettant de mettre fin à une activité en spécifiant ses états de sortie

5.1.2.2 Règles assistant l'apprenant dans l'activité

Les règles assistant les apprenants durant une activité permettent une assistance technique et pédagogique. Ainsi, ces règles permettent de guider les apprenants vers les ressources informatiques qui leur conviennent, d'explicitier l'activité proposée ou d'assister l'apprenant dans la réalisation de l'activité.

Ces règles utilisent des actions très variées qui dépendent de l'objectif du concepteur de l'assistance. Par exemple, pour guider les apprenants vers certaines ressources informatiques, les actions de type message et mise en valeur permettent de guider pas-à-pas les apprenants pour qu'ils atteignent ces ressources, alors que les actions d'assistance de type « action automatisée » et « ressources » dirigent les apprenants directement vers ces ressources.

Lors du lancement d'une activité, certaines règles associées à cette activité doivent être lancées. Par exemple, lors du lancement de l'activité, une règle est lancée pour guider les apprenants vers les ressources informatiques nécessaires à son apprentissage. Cependant, au moment de la définition de telles règles, le concepteur d'assistance ne sait pas quel événement va lancer des activités qu'il organise dans un guidage pédagogique (par exemple, un événement de type fin d'une activité avec des états de sortie). Nous avons donc ajouté et positionné dans la typologie des éléments aLDEAS un nouveau type d'événement « **lancement de l'activité** », facile à comprendre et à utiliser par les enseignants (cf. Annexe C page 212).

5.1.2.3 Mécanisme d'activation / désactivation des règles associées à l'activité

Plusieurs activités peuvent être associées à plusieurs règles. Certaines règles associées à différentes activités débutent avec un même événement déclencheur. Lorsque cet événement intervient, ces règles sont toutes lancées, même si elles sont associées à des activités différentes. Par exemple, une règle R_1 dans l'activité A_1 va mettre en valeur l'exercice 1 lors d'un clic sur la liste d'exercices. De même, une règle R_2 dans l'activité A_2 va mettre en valeur l'exercice 2 lors d'un clic sur la liste d'exercices. Si l'apprenant choisit de réaliser l'activité A_1 et s'il clique sur la liste d'exercices, il faut que le système de guidage mette en valeur l'exercice 1. Cependant, la règle R_2 attend également ce clic. L'exercice 2 est donc également mis en valeur. Pour régler ce problème, l'enseignant doit ajouter une condition qui vérifie quelle activité l'apprenant est en train de réaliser, ce qui demande du temps. De plus, pour la phase d'exécution d'un système d'assistance, le parcours de toutes les règles, et la vérification de l'activité en cours diminue les performances de SEPIA. Nous avons donc ajouté un nouveau mécanisme pour cette phase : **l'activation / désactivation des règles associées à une activité.**

Dans la phase d'exécution d'un système d'assistance, ce mécanisme permet d'activer seulement les règles qui sont associées à l'activité en cours, au lieu de toutes les règles comme c'était le cas avant. Il n'y a donc que les règles activées qui peuvent être lancées par le moteur d'assistance lors d'un événement détecté. Pour l'activation et la désactivation des règles, il faut respecter trois principes :

- Les règles qui ne sont associées à aucune activité sont toujours activées.
- Les règles qui sont associées à une activité sont activées quand l'apprenant est en train de réaliser cette activité.
- Les règles qui sont associées à une activité sont désactivées quand l'apprenant est en train de réaliser une autre activité.

5.2. Modèle de guidage pédagogique

Le modèle de guidage pédagogique explicite comment représenter les cinq types de guidage pédagogique que nous avons identifiés dans notre état de l'art et comment les définir sous forme de règles aLDEAS compatibles avec le moteur d'assistance de SEPIA. Chaque type est représenté par sa représentation graphique dans SEPIA-edu et sa définition en règles aLDEAS. Dans ce modèle, les n activités contenues dans un guidage pédagogique sont nommées A_i , avec $i \in [1, n]$.

5.2.1 Guidage Libre

Dans le guidage libre, toutes les activités sont proposées en même temps à l'apprenant. Il n'y a aucune contrainte sur l'ordre de proposition et sur la date à laquelle ces activités sont disponibles. Les apprenants peuvent donc accéder librement à toutes les activités contenues dans le guidage.

La représentation d'un guidage libre doit être conforme à la partie haute de la Figure 115. Ce guidage débute par les événements déclencheurs. Le guidage libre est représenté par un ensemble d'activités placées au même niveau sans aucune contrainte.

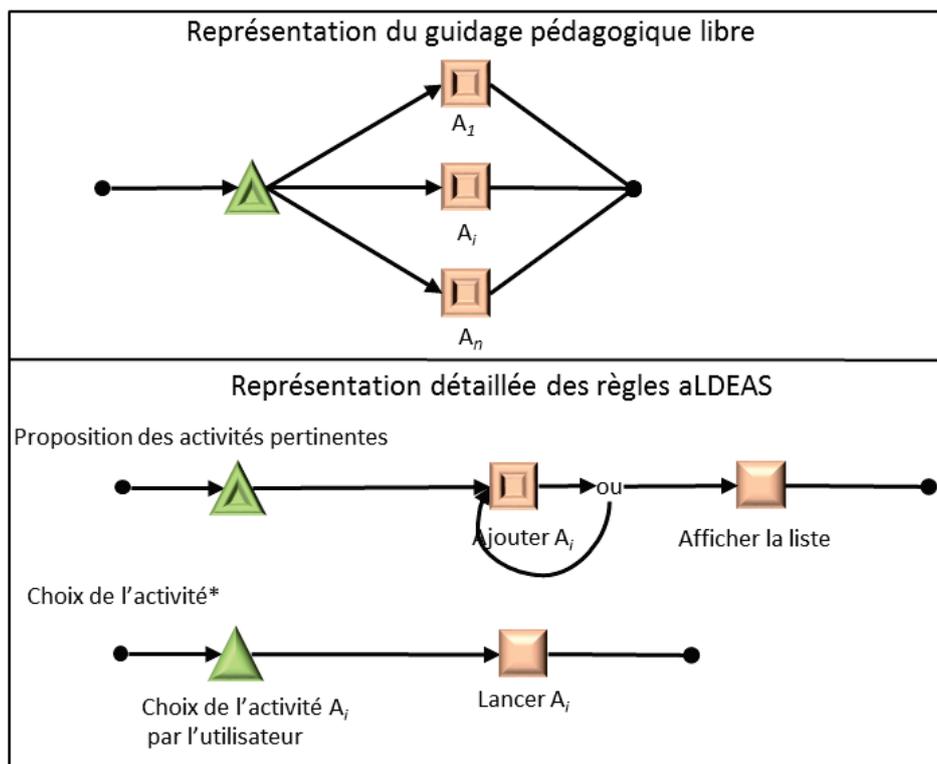


Figure 115 : Modèle de guidage libre

La partie basse de la Figure 115 montre les règles qui définissent un guidage libre. Lors du lancement du guidage, la règle « proposition d'activités pertinentes » est lancée. Pour cela, cette règle débute par les événements déclencheurs qui lancent le guidage. Elle ajoute toutes les activités A_i dans une liste d'activités et affiche cette liste en utilisant le type d'actions d'assistance « message dynamique ».

La règle « proposition d'activités pertinentes » permet d'afficher les activités pertinentes sous forme d'une liste d'activités. Cependant, connaître le choix de l'activité fait par l'apprenant nécessite des règles « choix d'une activité ». Une telle règle débute par le choix d'une activité A_i par l'apprenant. Il s'agit d'un clic de l'apprenant sur une activité dans la liste d'activités proposée dans le message créé dynamiquement. Suite à cet événement, la règle lance l'activité A_i .

5.2.2 Guidage séquentiel

Dans le guidage séquentiel, les activités sont proposées les unes après les autres. La première activité est proposée à l'apprenant. S'il termine cette activité avec succès, l'activité suivante lui est proposée.

La représentation d'un guidage séquentiel doit être conforme à la partie haute de la Figure 116. Les activités sont ordonnées selon l'ordre dans lequel les activités seront proposées aux apprenants.

La partie basse de la Figure 116 illustre les règles qui définissent un guidage pédagogique. La première règle R_1 est proposée lors du lancement du guidage pédagogique. Elle débute par les événements déclencheurs qui lancent ce guidage. Elle lance la première activité dans la liste

d'activités séquentielles. Lors de la fin d'une activité A_i ($i \in [1, n-1]$), l'activité suivante A_{i+1} est proposée à l'apprenant. Pour cela, les règles R_i ($i \in [2, n]$) débutent par un événement déclencheur « fin A_{i-1} ». La règle R_i lance l'activité suivante A_i .

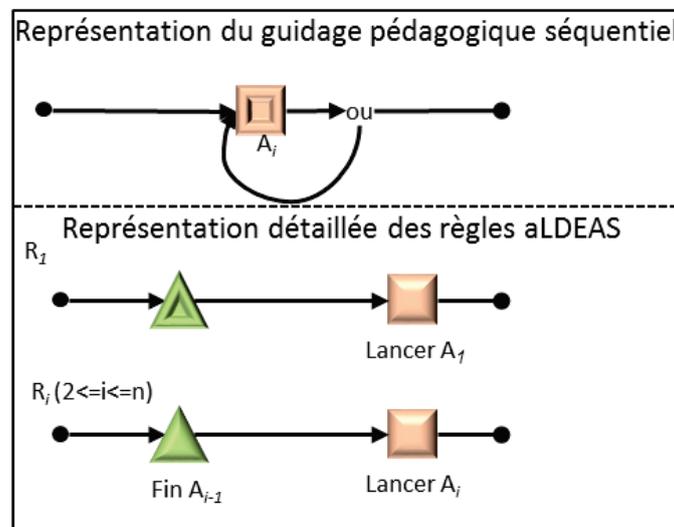


Figure 116 : Modèle de guidage séquentiel

5.2.3 Guidage temporel

Dans le guidage temporel, les activités sont proposées temporellement. Autrement dit, la proposition des activités dépend du moment où l'apprenant demande une activité.

La représentation d'un guidage temporel doit être conforme à la partie haute de la Figure 117. Ce guidage débute par les événements déclencheurs et vérifie le temps actuel par une condition. Chaque activité est associée à une alternative avec une valeur de temps.

La partie basse de la Figure 117 illustre les règles qui définissent un guidage pédagogique temporel. Ces règles ressemblent à celle du guidage libre. Ainsi, les règles « proposition d'activités pertinentes » et « choix de l'activité » sont nécessaires. Cependant, la première règle n'ajoute pas directement les activités dans une liste d'activités à afficher, mais elle lance une règle « vérification de l'activité ». Cette règle débute par un événement déclencheur « lancement par une règle » et vérifie la contrainte temporelle. Il s'agit d'une consultation du temps actuel. Pour une activité A_i , si le temps actuel est compatible avec la valeur t_i , la règle ajoute l'activité A_i dans la liste d'activités.

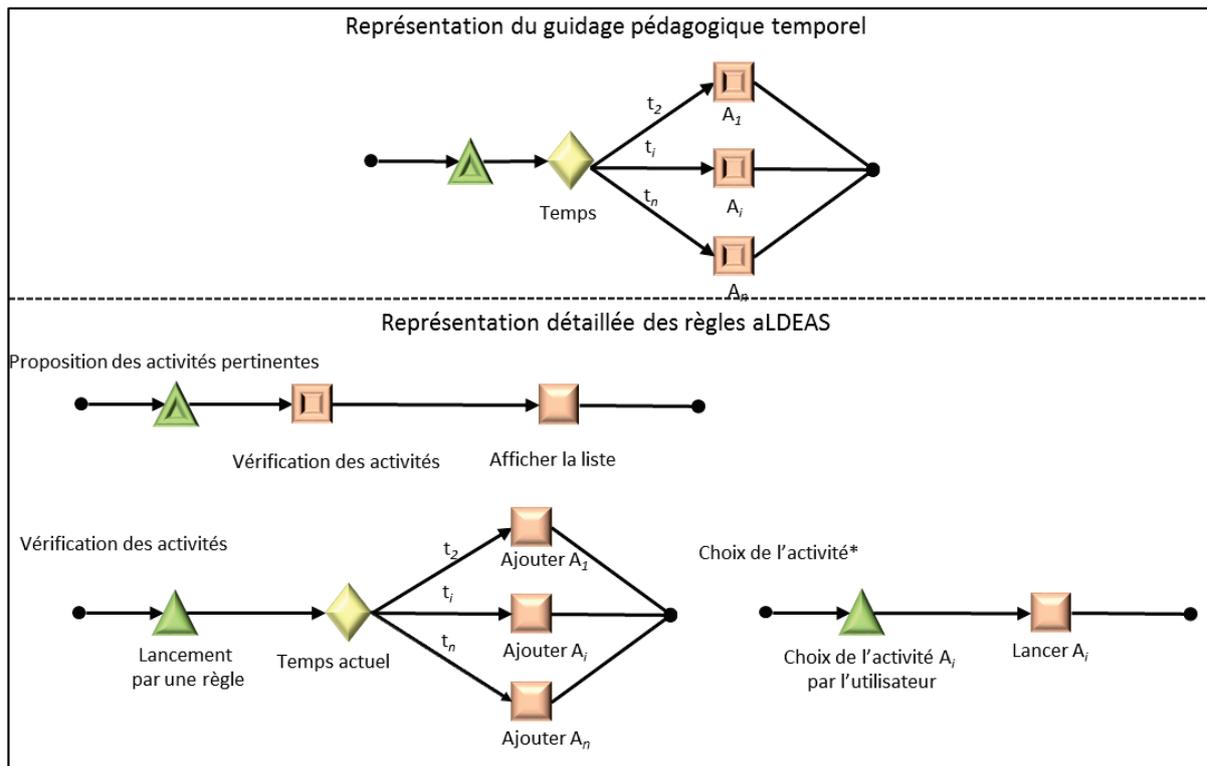


Figure 117 : Modèle de guidage temporel

5.2.4 Guidage personnalisé

Dans le guidage personnalisé, les activités sont proposées en fonction du profil de l'apprenant. Autrement dit, la proposition des activités dépend des informations personnelles de l'apprenant contenues dans son profil ou des informations concernant les états de sortie d'une activité (qui sont également stockées dans le profil de l'apprenant).

La représentation d'un guidage temporel doit être conforme à la partie haute de la Figure 118. Ce guidage débute par les événements déclencheurs et vérifie une condition du profil ou des états de sortie de l'activité. Chaque activité est associée à une alternative avec une valeur (textuelle, numérique, etc.).

La partie basse de la Figure 118 illustre les règles qui définissent un guidage pédagogique personnalisé. Ces règles ressemblent à celles du guidage temporel. Ainsi, les règles « proposition d'activités pertinentes », « vérification des activités » et « choix de l'activité » sont nécessaires. Cependant, dans le guidage personnalisé, la règle « vérification des activités » consulte le profil de l'apprenant au lieu du temps actuel.

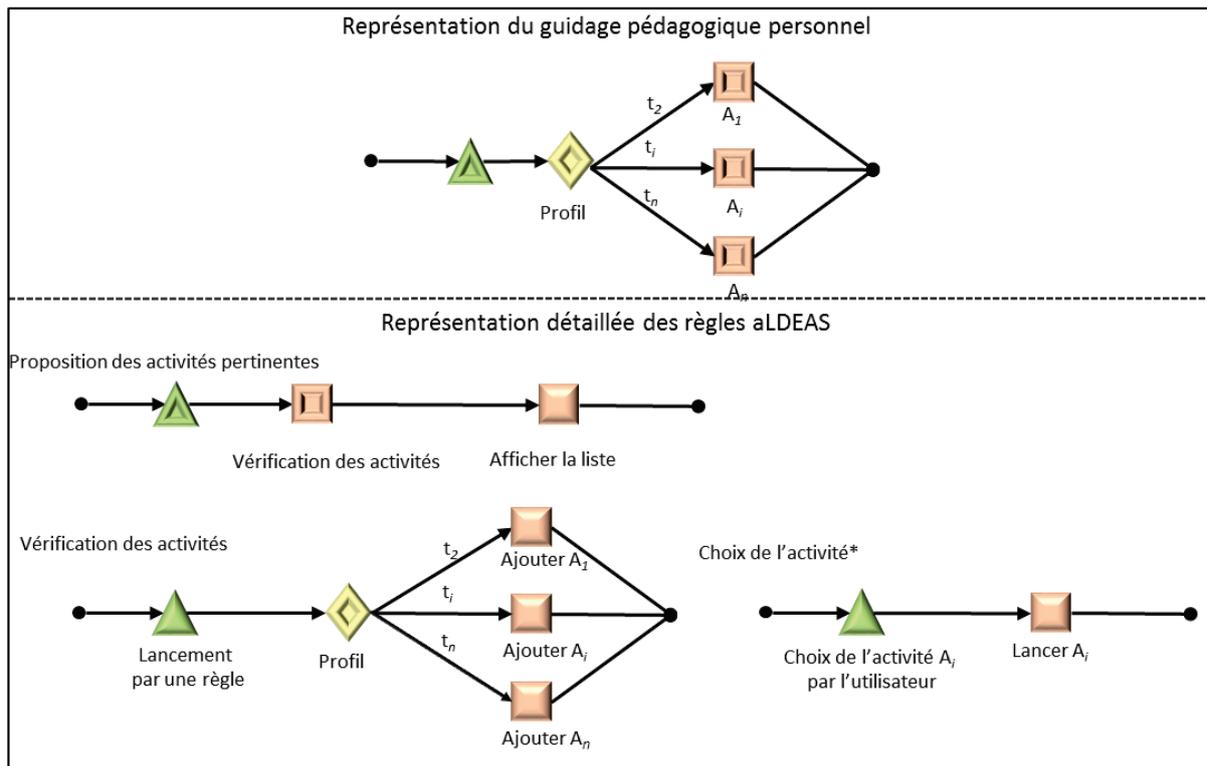


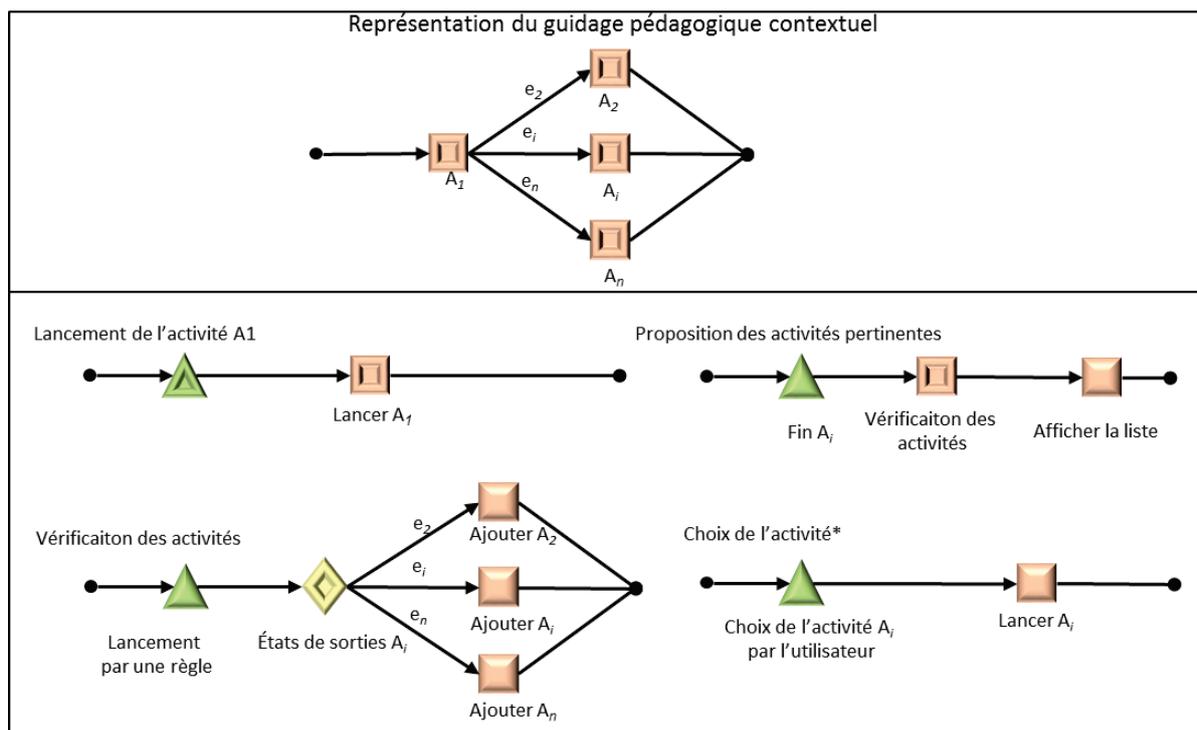
Figure 118 : Modèle de guidage personnalisé

5.2.5 Guidage contextuel

Dans le guidage contextuel, les états d'une activité contraignent la proposition des activités suivantes. Ils peuvent donc donner lieu à la proposition d'activités de remédiation.

La représentation d'un guidage contextuel doit être conforme à la partie haute de la Figure 119. Ce guidage débute par une activité. En fonction des états de sortie de cette activité, les activités suivantes A_i ($i \geq 2$) sont proposées. Chaque activité A_i ($i \geq 2$) est associée à une alternative selon la valeur des états de sortie e_i .

La partie basse de la Figure 119 illustre les règles qui définissent un guidage contextuel. La règle « lancement de l'activité A_1 » qui lance la première activité est nécessaire. Les autres règles sont presque semblables à celles du guidage temporel. Ainsi, les règles « proposition d'activités pertinentes », « vérification des activités » et « choix de l'activité » sont nécessaires. Cependant, dans le guidage contextuel, la règle « proposition d'activités pertinentes » débute par un événement déclencheur « fin A_1 » au lieu du lancement du guidage. Elle lance la règle « vérification des activités » qui consulte les états de sortie de l'activité A_1 pour ajouter les activités suivantes pertinentes dans la liste d'activités.



5.3. Patron de guidage pédagogique

Si le modèle de guidage pédagogique permet de définir les cinq types de guidage pédagogique que nous avons identifiés, il n'explique pas comment combiner ces types. Nous avons donc proposé un patron de guidage pédagogique qui explicite comment définir un guidage pédagogique complet qui peut cumuler les cinq types de guidages pédagogiques que nous venons de présenter. Il permet de plus de définir comment ces différents guidages vont être transcrits en règles aLDEAS compatibles avec SEPIA-edu. Ce patron permet de définir facilement et explicitement un guidage pédagogique complet contenant un des cinq types de guidage mais également des combinaisons de ceux-ci.

Dans ce patron (cf. Figure 120), un guidage débute par les événements déclencheurs. Ces événements peuvent être issus du lancement par d'autres règles. Une règle peut donc lancer un guidage pédagogique comme elle peut lancer une action, une règle, ou un bloc. Le guidage est défini par l'organisation d'un ensemble d'activités. La proposition d'une activité peut être contrainte par plusieurs sources : une condition temporelle, une condition sur le profil ou une condition sur les états de sortie. Cela correspond aux types de guidage pédagogique temporel, personnalisé et contextuel.

De plus, une activité peut être proposée directement ou après une ou plusieurs activités. De même, une activité peut être suivie par aucune, une ou plusieurs activités.

Dans le cas où une activité n'est suivie par aucune activité, elle se termine par la fin du bloc. Dans le cas où une activité est suivie par plusieurs activités, il s'agit d'un guidage séquentiel ou contextuel. La flèche qui sort de cette activité porte les valeurs des états de sortie ou non. Ces états contraignent la proposition de l'activité suivante. Cette flèche peut aller vers une

condition ou non. Cette condition permet de combiner les guidages de types temporel ou personnalisé avec les guidages de types séquentiel et contextuel.

L'activité suivante peut suivre une ou plusieurs activités précédentes. Il faut donc une condition « et/ou » entre ces activités précédentes. La condition « et » signifie que l'activité suivante est proposée quand les conditions sur les états de sortie des activités précédentes sont toutes valides. La condition « ou » signifie que l'activité suivante est proposée quand au moins une condition sur les états de sortie d'une activité parmi les activités précédentes est valide. Dans le cas où une activité est suivie par une seule activité, la condition « et/ou » n'apparaît pas.

En particulier, il se peut qu'une activité soit proposée après des activités et/ou si une condition quelconque est valide. Il s'agit du cas du chemin bas de la Figure 120, avec une seule condition. Dans les cas précédents, les conditions sont placées juste avant les activités pour assurer que ces activités soient toujours contraintes par ces conditions. Dans le dernier cas, l'activité n'est pas toujours contrainte par la condition.

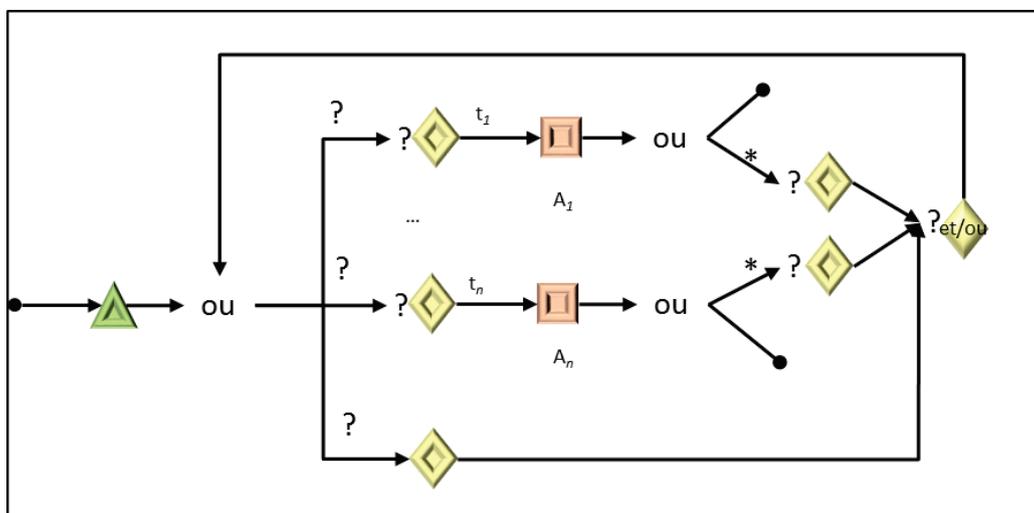


Figure 120 : Patron de guidage pédagogique

Pour illustrer l'utilisation de ce patron, prenons l'exemple du site Web Matou Matheux. Plus précisément, nous souhaitons créer un guidage pédagogique sur un ensemble d'exercices portant sur la comparaison des quantités. Cet exemple est également exploité dans la section suivante pour illustrer le processus de transformation d'un guidage pédagogique en règles aLDEAS.

Ce site fournit six exercices qui sont regroupés en deux catégories : « moins de 10 » et « plus de 10 » représentant les exercices faisant intervenir des comparaisons de nombres dont l'écart est respectivement inférieur ou supérieur à 10. Nous créons donc en aLDEAS les six activités correspondantes : d'une part « les animaux » (nommée A_1), « les dominos » (nommée A_2) et « les euros » (nommée A_3) pour la catégorie « moins de 10 » et d'autre part « les jouets » (nommée A_4), « les dominos chinois » (nommée A_5) et « les centimes d'euros » (nommée A_6) pour la catégorie « plus de 10 ».

Ce guidage pédagogique est fourni aux apprenants lors du lancement du système d'assistance. Dans ce guidage, les activités ne sont disponibles que pour les apprenants de

classe de CP. De plus, les activités pour « moins de 10 » sont disponibles entre le 01/10/2016 et le 31/10/2016, alors que les activités pour « plus de 10 » sont accessibles à partir du 01/11/2016. Les activités pour chaque catégorie sont proposées l'une après l'autre par difficulté croissante, c'est-à-dire selon l'ordre suivant : A₁, A₂, A₃ pour la catégorie d'activités « moins de 10 » et A₄, A₅, A₆ pour la catégorie d'activités « plus de 10 ». En cas d'échec à une activité A₄, A₅ ou A₆, l'assistance propose comme activité de remédiation respectivement les activités A₁, A₂ ou A₃.

Le guidage que nous souhaitons créer est un guidage qui cumule plusieurs types de guidage pédagogique. Pour définir ce guidage, nous utilisons le modèle d'activité, le modèle de guidage pédagogique et le patron associé.

Tout d'abord, le modèle d'activité nous permet d'identifier six activités A₁, A₂, A₃, A₄, A₅, A₆ avec pour états de sortie respectifs S₁, S₂, S₃, S₄, S₅, S₆ dont l'échelle est le succès de l'activité avec deux valeurs possibles : succès et échec.

Le patron de guidage pédagogique permet de définir le guidage pédagogique. La Figure 121 illustre le guidage de notre exemple ainsi défini. Les activités sont contraintes par les conditions C₁ et C₂. Ces conditions sont une combinaison logique « et » des conditions de temps et de profil. La condition du temps impose que les activités soient accessibles seulement pendant une période précise (du 01/10/2016 au 31/10/2016 pour C₁ et à partir du 01/11/2016 pour C₂), alors que la condition du profil impose qu'elles soient accessibles pour le niveau CP dans les deux cas. Les alternatives associées à C₁ et C₂ portent la valeur « vrai ». T est le temps actuel et N est le niveau de l'apprenant dans le profil. La contrainte C₁ est donc égale à (T ∈ [01/10/2016, 31/10/2016]) && (N = « CP ») et la C₂ à (T >= 1/11/2016) && (N = « CP »).

Pour que les activités A₁, A₂, A₃ et les activités A₄, A₅, A₆ soient planifiées séquentiellement, les liens entre ces activités comportent un état de sortie avec la valeur « succès ». Un échec à l'activité A₄, A₅ ou A₆ donne lieu à la proposition d'activités de remédiation respectivement A₁, A₂ ou A₃. Les liens entre les activités portent donc sur la valeur « échec » de l'état de sortie de ces activités. Comme l'activité A₁ (respectivement A₂, A₃) est proposée si la condition C₁ est valide **ou** si l'activité A₄ (respectivement A₅, A₆) se termine par un échec, un « ou » entre la condition C₁ et l'activité A₄ (respectivement A₅, A₆) est nécessaire.

Dans le cas où des conditions qui contraignent une activité sont fausses, cette activité n'est pas proposée à l'apprenant, c'est-à-dire qu'elle n'est pas présente dans la liste d'activités.

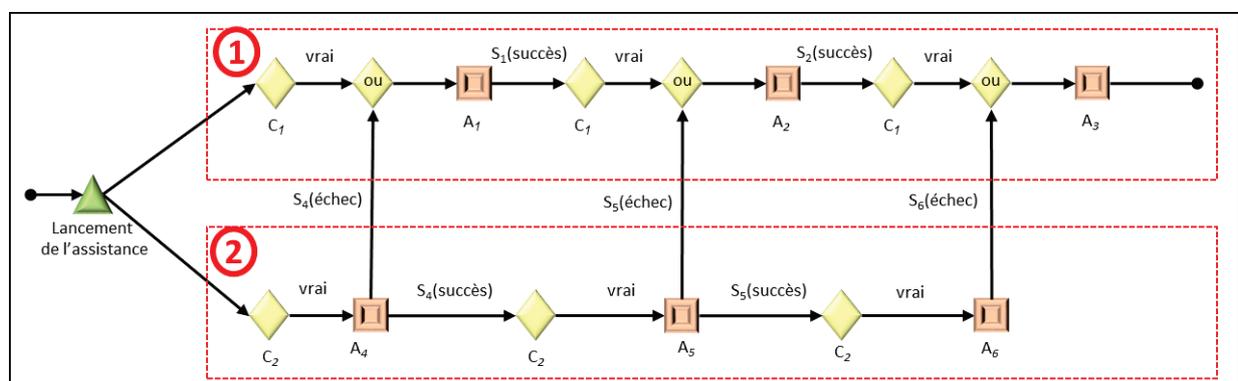


Figure 121 : Exemple de guidage pédagogique défini avec le patron de guidage pédagogique

Ce patron de guidage pédagogique permet aux enseignants de mettre en œuvre des guidages complets. Cependant, le guidage défini avec ce patron n'était pas possible avec la version précédente de SEPIA. Pour qu'il puisse être exécuté par le moteur de SEPIA, il faut traduire le guidage pédagogique en règles aLDEAS. Nous proposons donc la traduction du guidage *via* un ensemble de règles aLDEAS. Dans la Figure 122, nous regroupons les règles en quatre catégories : proposition d'activités pertinentes, vérification de l'activité, choix de l'activité et fin de l'activité.

Lors du lancement du guidage, la règle « proposition d'activités pertinentes » est lancée. Cette règle débute par les mêmes événements déclencheurs que le guidage pédagogique. Ainsi, cette règle est lancée lors du lancement du guidage pédagogique. Elle lance les règles de vérification des activités A_i pour ajouter les règles valides dans une liste d'activités. Enfin, elle affiche cette liste en utilisant un type d'actions d'assistance « afficher un message dynamique ».

Les règles de vérification d'une activité A_i « vérification de l'activité » débutent par le lancement de la règle « proposition d'activités pertinentes ». Cette règle vérifie toutes les contraintes qui sont une combinaison des conditions « état de sortie », « profil » et « temps ». Si cette combinaison est valide, la règle ajoute l'activité A_i dans la liste d'activités.

Les règles « proposition d'activités pertinentes » et « vérification de l'activité » permettent d'afficher les activités pertinentes grâce à une liste d'activités. Cependant, pour pouvoir connaître le choix d'activité de l'apprenant, les règles « choix de l'activité » sont nécessaires. Une telle règle débute par le choix d'une activité A_i par l'apprenant, qui se fait par un clic de l'apprenant sur une activité de la liste. Lors de cet événement, la règle lance l'activité A_i .

Lors de la fin de l'activité, le guidage est terminé. Pour maintenir le déroulement d'un guidage pédagogique même si une activité est terminée, il faut révérifier les activités et les afficher aux apprenants. C'est la règle « fin des activités » qui est chargée de le faire. Cette règle débute par un ensemble d'événements déclencheurs « fin de l'activité A_i ». Lors de la fin d'une activité A_i , elle lance les mêmes règles et actions que la règle « proposition d'activités pertinentes ». En pratique, les deux règles « fin des activités » et « proposition d'activités pertinentes » sont combinées en fusionnant les événements déclencheurs de ces règles.

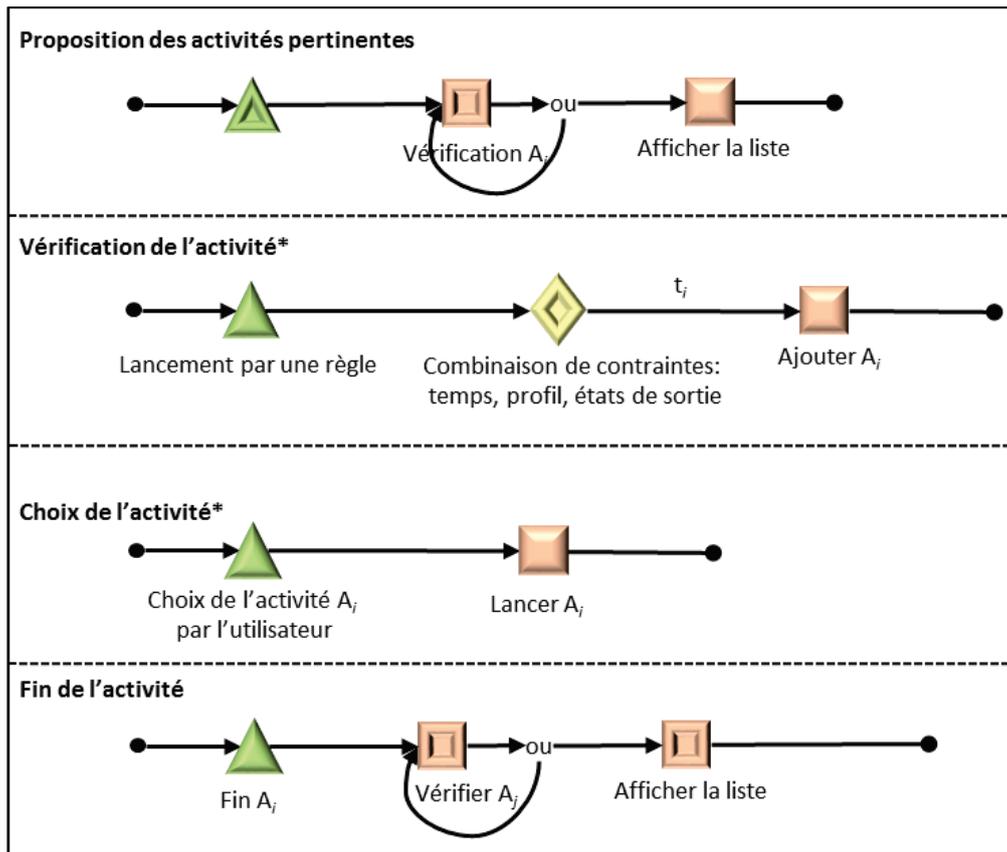


Figure 122 : Traduction du patron de guidage pédagogique en règles aLDEAS

5.4. Processus de transformation d'un guidage pédagogique en règles aLDEAS

Le patron de guidage pédagogique explicite les règles aLDEAS qui constituent un guidage pédagogique. Le processus de transformation du guidage pédagogique explique comment la transformation du guidage en règles aLDEAS est effectué. La Figure 123 illustre ce processus avec deux étapes principales : initialisation des listes et transformation du guidage pédagogique.

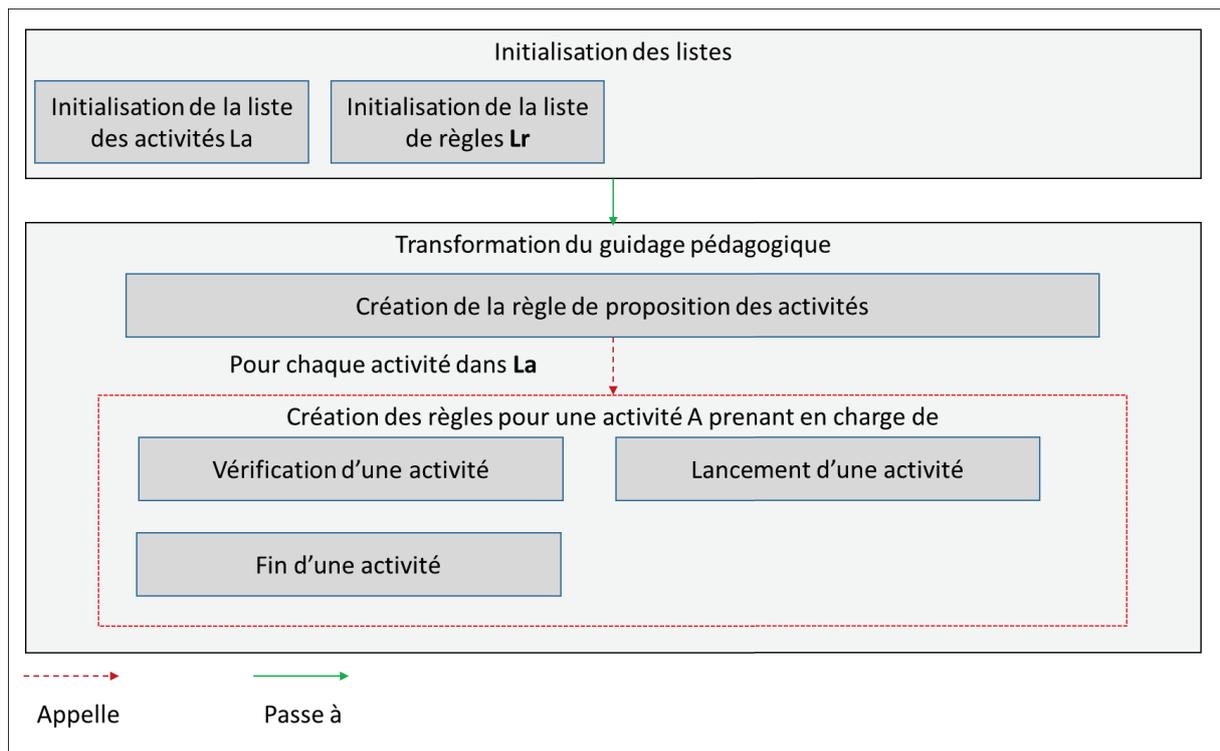


Figure 123 : Processus de transformation d'un guidage pédagogique en règles aLDEAS

5.4.1 Initialisation des listes

Dans cette étape, les activités d'un guidage pédagogique sont ajoutées dans la liste La. Une liste de règles Lr qui contiendra les règles générées est initialisée sans aucune règle.

Après l'initialisation des listes, la liste de règles Lr doit être complétée. C'est l'étape suivante de « Transformation du guidage pédagogique » qui s'en charge.

5.4.2 Transformation du guidage pédagogique

Cette étape a pour objectif la transformation d'un guidage pédagogique en règles aLDEAS. Cette étape commence par la création de la règle « proposition d'activités pertinentes » appelée R_x . R_x doit débiter par les événements déclencheurs du guidage pédagogique. Dans le cas où l'événement déclencheur est le lancement par les autres règles. Le guidage pédagogique dans ces règles est transformé en R_x . Afin de compléter R_x , une procédure « création des règles pour une activité » est appelée pour chaque activité dans le guidage.

Pour une activité A, dans cette procédure, la sous-étape « vérification de l'activité » permet de créer la règle « vérification de l'activité ». Cette règle contient une combinaison de conditions issues des contraintes sur l'activité en question. Après sa création, la règle R_x est également mise à jour. Ainsi, la règle R_x doit contenir la règle « vérification de l'activité » pour pouvoir lancer l'activité. La sous-étape « lancement d'une activité » permet de créer la règle « choix de l'activité ». Dans la sous-étape « fin d'une activité », la règle « fin des activités » n'est pas créée, mais elle est fusionnée avec R_x . Ainsi, cette sous-étape permet d'ajouter un événement déclencheur « fin de A » dans R_x .

L'étape « transformation du guidage pédagogique » est terminée si la procédure est terminée pour toutes les activités du guidage. La liste L_r est complétée avec les règles générées. L'ensemble des règles générées et des règles définies par les enseignants constituent un système d'assistance complet contenant le guidage pédagogique traduit en règles aLDEAS.

Les algorithmes de cette étape sont donnés dans l'annexe D.2 page 220.

Dans la section précédente, le guidage pédagogique de l'exemple Matou Matheux est défini avec le patron de guidage pédagogique. Pour qu'il puisse être défini complètement en règles aLDEAS, nous appliquons étape par étape le processus de transformation d'un guidage pédagogique en règles aLDEAS.

Étape d'initialisation des listes

L_a : $A_1, A_2, A_3, A_4, A_5, A_6$

L_r : pas de règles générées

Étape de transformation du guidage pédagogique

Création de la règle de proposition des activités

Cette étape permet de créer une règle R_0 (cf. Figure 124). Cette règle débute par l'événement déclencheur « lancement de l'assistance ». Dans cette étape, la règle ne contient qu'une action à lancer « afficher la liste ».

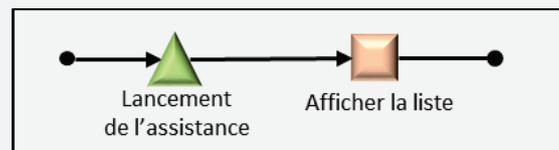


Figure 124 : Règle R_0 générée dans l'étape de « création de la règle de proposition des activités »

Pour chaque activité de la liste L_a , nous appliquons la procédure « création des règles pour une activité ».

Création des règles pour une activité

Pour l'activité A_1 , une règle « vérification de l'activité » est créée. Cette règle est lancée par la règle R_0 qui contient une combinaison de conditions. Selon le guidage défini par le patron de guidage pédagogique, l'activité est contrainte par une condition C_1 ou par l'échec de l'activité A_4 : S_4 (échec). Donc, une combinaison de conditions « $C_1 | S_4$ (échec) » est créée. La règle R_{11} va ajouter l'activité A_1 dans la liste d'activités à afficher quand cette combinaison est vraie. Ensuite, une règle R_{12} « choix de l'activité » est créée. Elle débute par le choix de l'activité A_1 pour lancer cette activité. Enfin, la règle R_0 est mise à jour. Un événement déclencheur et la règle R_{12} y sont ajoutés.

Après avoir appliqué cette procédure pour A_1 , la liste de règles est mise à jour : R_0 (mise à jour), R_{11} (créée), R_{12} (créée) (cf. Figure 125).

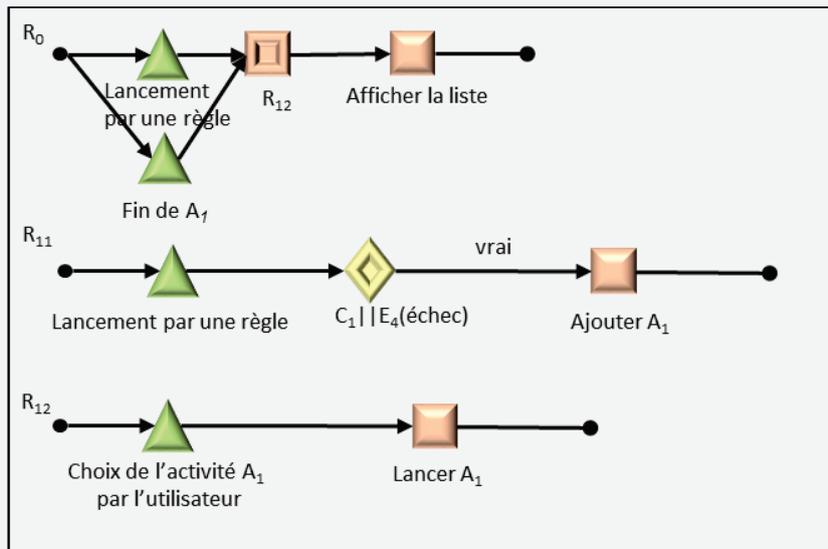


Figure 125 : Règle R_0 mise à jour et règles R_{11} et R_{12} générées

Le même processus est appliqué pour l'activité A_2 . Il faut créer les règles R_{21} et R_{22} et mettre à jour la règle R_0 . Dans la règle R_{21} « vérifier l'activité », la combinaison est plus complexe. D'une part, l'activité A_2 est contrainte par le succès de l'activité A_1 : S_1 (succès) et une condition C_1 , et d'autre part, l'activité A_2 est contrainte par l'échec de l'activité A_5 : S_5 (échec). La combinaison pour la vérification est donc une combinaison logique : $(C_1 \& \& E_1(\text{succès})) \mid \mid E_5(\text{échec})$.

Après avoir appliqué cette procédure pour A_2 , la liste des règles est mise à jour et contient R_0 (mise à jour), R_{11} (inchangée), R_{12} (inchangée), R_{21} (créée) et R_{22} (créée) (cf. Figure 126).

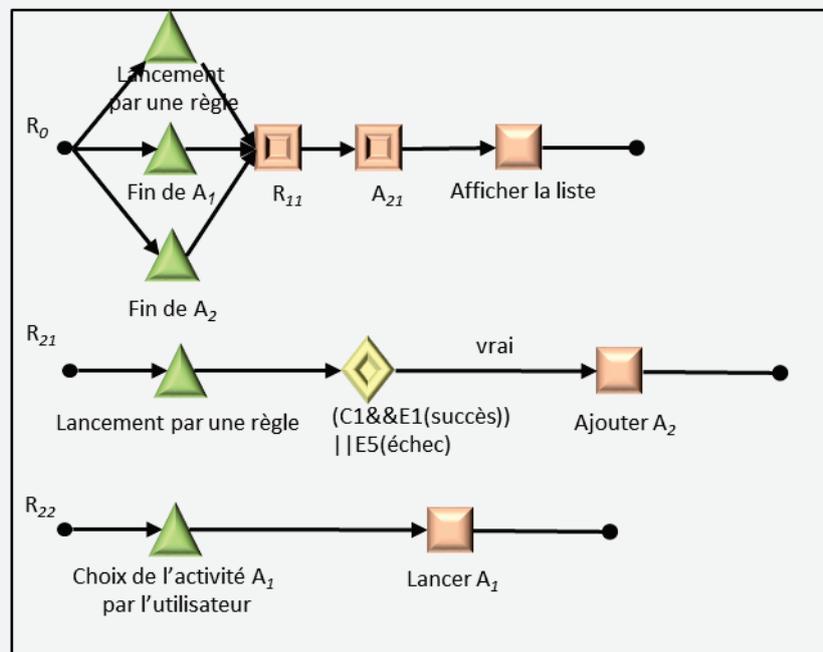


Figure 126 : Règle R_0 mise à jour et règles R_{21} , R_{22} générées

Après avoir appliqué cette procédure pour toutes les activités, nous avons une liste de règles générées R_0 , R_{11} , R_{12} , R_{21} , R_{22} , R_{31} , R_{32} , R_{41} , R_{42} , R_{51} , R_{52} , R_{61} , R_{62} .

5.5. SEPIA-edu : mise en œuvre du guidage pédagogique

Dans la version précédente issue de la thèse de Blandine Ginon (Ginon, 2014), l'éditeur d'assistance de SEPIA permettait aux enseignants de définir un guidage pédagogique sous forme de règles d'assistance (cf. Figure 93). Ces règles étaient listées dans un tableau de règles. Il ne proposait pas les concepts d'« activité » et de « guidage pédagogique ». Les enseignants devaient créer des règles pour définir implicitement ces concepts.

En utilisant nos modèles, les enseignants peuvent à présent définir un guidage pédagogique sous forme règles aLDEAS. Cependant, cette tâche reste complexe. Pour soutenir explicitement la définition du guidage pédagogique, nous avons fait évoluer l'éditeur de SEPIA-edu en implémentant notre modèle d'activité, notre patron de guidage pédagogique et notre processus de transformation du guidage en règles aLDEAS.

Le modèle d'activité permet d'identifier les activités d'apprentissage sur les applications-cibles. Le patron de guidage pédagogique permet de définir un guidage pédagogique dans lequel les activités identifiées s'organisent en fonction des besoins des enseignants. Leur implémentation (modèle d'activité et patron de guidage pédagogique) dans SEPIA-edu donne lieu à l'ajout des étapes d'« identification des activités » et de « définition du guidage pédagogique » qui s'adressent aux enseignants (cf. Figure 127).

Cependant, l'activité et le guidage pédagogique ne sont pas exécutables directement par le moteur de SEPIA-edu dont le fonctionnement s'appuie sur l'exécution des règles. Pour maintenir le fonctionnement préalable du moteur de SEPIA, nous avons implémenté le processus de transformation d'un guidage pédagogique en règles aLDEAS pour ajouter une étape à l'éditeur de SEPIA-edu « transformation du guidage pédagogique en règles aLDEAS ». Cette étape est automatisée. De plus, pour avoir une solution complète, nous avons implémenté le nouveau type d'actions d'assistance « message interactif dynamique » qui permet d'afficher la liste d'activités, le nouveau type de consultations « états de sortie » qui permet de consulter les valeurs des états de sortie et le mécanisme d'« activation/désactivation des règles ».

En implémentant nos modèles, nous avons fait évoluer l'architecture de SEPIA (cf. Figure 127) pour expliciter et faciliter la définition du guidage pédagogique et son exécution. Dans cette architecture, les composants en bleu sont les composants de la version initiale de SEPIA. Les composants en vert sont les composants ajoutés dans SEPIA-edu. Enfin, les composants en orangé sont les composants modifiés. Nous revenons à présent sur chacun des composants verts et orangés pour expliquer leur fonctionnement.

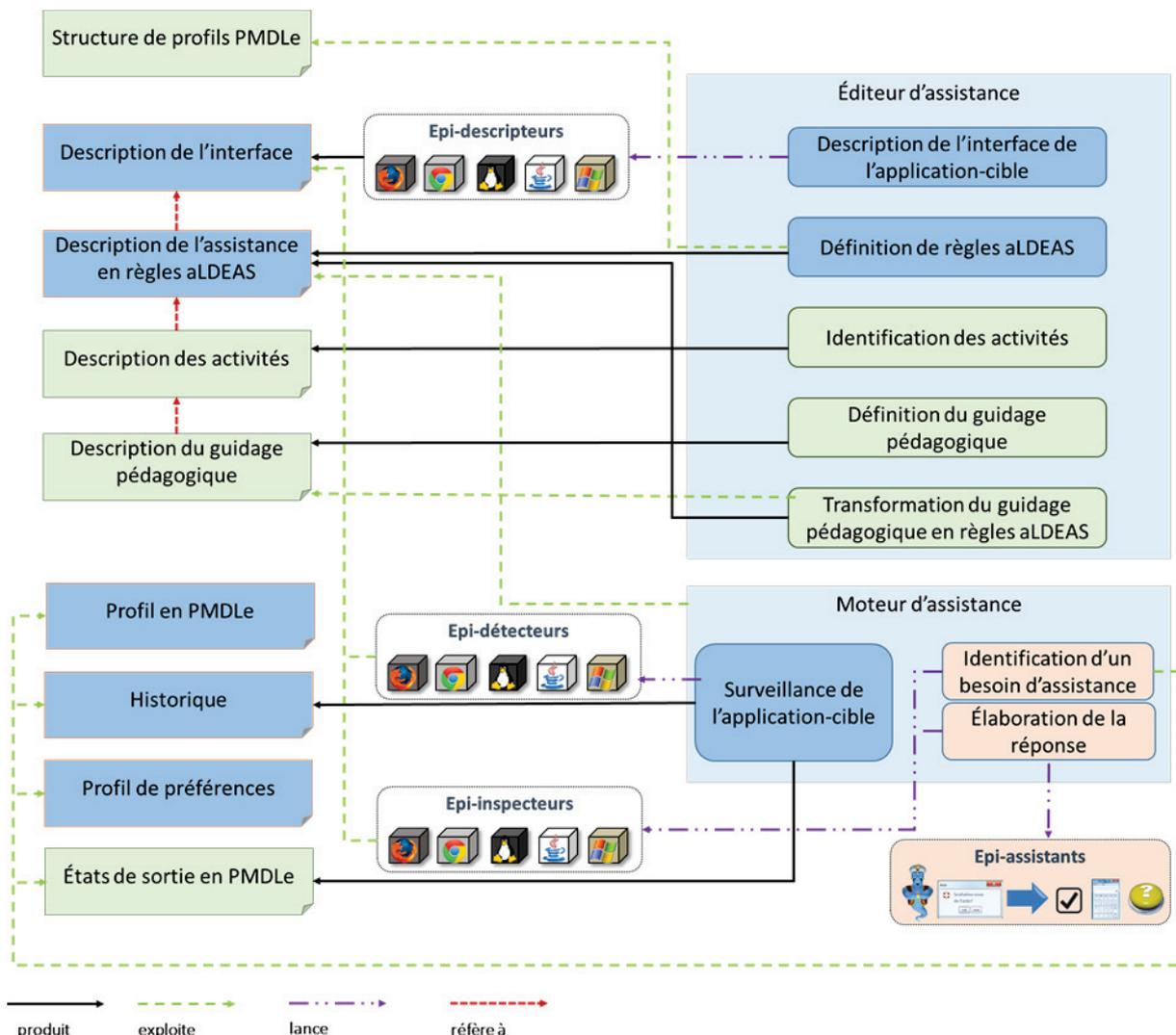


Figure 127 : Architecture de l'éditeur de SEPIA-edu adoptant le modèle d'activité, le patron de guidage pédagogique et le processus de transformation du guidage pédagogique en règles aLDEAS.

5.5.1 Identification des activités

En implémentant le modèle d'activité, l'identification des activités est intégrée dans l'éditeur d'assistance de SEPIA-edu (cf. Figure 127). Cette étape est dédiée au concepteur d'assistance. Elle lui permet d'identifier les activités organisées dans un guidage pédagogique.

Cette étape est réalisée après l'étape de description de l'interface de l'application-cible mais avant, après ou pendant l'étape de définition des règles aLDEAS.

Dans l'implémentation du modèle d'activité, l'identification d'une activité comporte trois sous-étapes principales : la déclaration d'une activité, la définition des états de sortie et l'association des règles aux activités. Le résultat produit dans cette étape est la description des activités exploitée pour définir un guidage pédagogique.

5.5.1.1 Déclaration d'une activité

Cette étape permet aux enseignants de déclarer une activité sur l'application-cible. Pour cela, l'enseignant doit préciser les informations générales de l'activité, c'est-à-dire son nom et sa description. Par exemple, l'activité d'identifiant T3 a pour nom « Les euros (<10) » et pour description « la comparaison des euros » (cf. A dans la Figure 128).

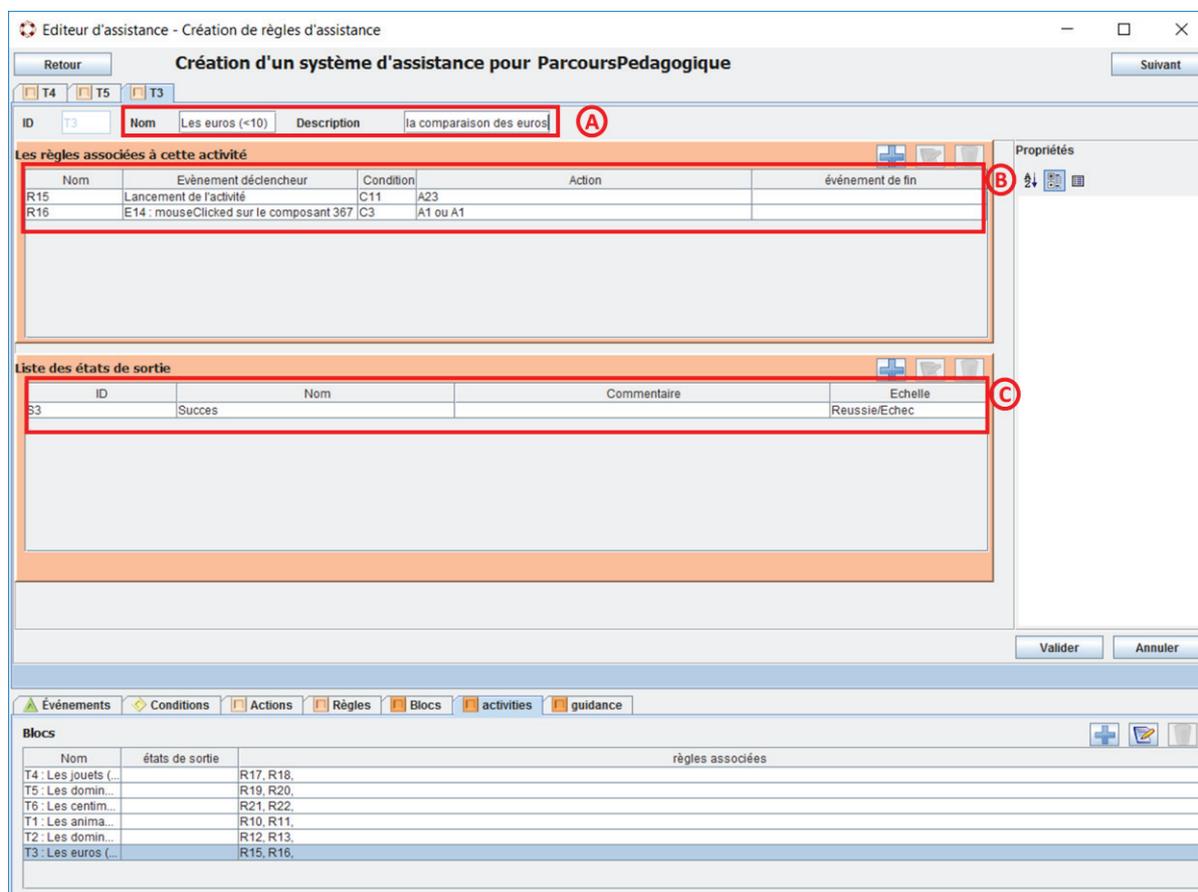


Figure 128 : Identification d'une activité dans SEPIA-edu

5.5.1.2 Définition des états de sortie

Après la déclaration d'une activité, l'enseignant définit les états de sortie associés possibles. Pour permettre cette définition, une interface a été implémentée. Les états de sortie sont formalisés en PMDL. Ils sont définis par un nom, une description et une échelle. La Figure 129 montre l'écran de définition d'un état de sortie avec une échelle de type « réussite » pour la réalisation d'une activité.

ID:
 Nom:
 Commentaire:
 Échelle:

Figure 129 : Définition d'un état de sortie d'une activité dans SEPIA-edu

Pour la définition d'une échelle, l'éditeur permet deux types d'échelle : textuelle et numérique. La Figure 130 montre l'écran de définition d'une échelle. Sur cet écran, une échelle textuelle est définie avec trois valeurs qui représentent le niveau de maîtrise : « Non maîtrisé », « Partiellement maîtrisé » et « Maîtrisé ».

ID:
 Nom:
 Type:
 Valeurs:

Figure 130 : Définition d'une échelle dans SEPIA-edu respectant le langage PMDL

5.5.1.3 Association des règles

Pour que l'activité soit exécutable et assistée avec SEPIA-edu, il est nécessaire d'y associer des règles (cf. section 5.1.2). Cette association d'une règle à une activité est faite lors de la phase de définition de la règle. La Figure 131 montre la définition d'une règle que l'enseignant associe à l'activité « Les jouets (>10) » (cf. A dans la Figure 131). En général, une règle peut être associée à différentes activités pour faciliter sa réutilisation (par exemple, une règle peut servir à fournir un même indice lors du clic sur un même bouton d'aide par l'apprenant dans deux activités différentes). Cependant, la représentation de l'association d'une règle aux différentes activités pendant la définition de cette règle reste un peu complexe pour les enseignants. La version actuelle de l'éditeur de SEPIA-edu permet donc d'associer une règle à une seule activité. Nous souhaitons toutefois améliorer cette interface pour permettre la prise en compte de l'association d'une règle à différentes activités.

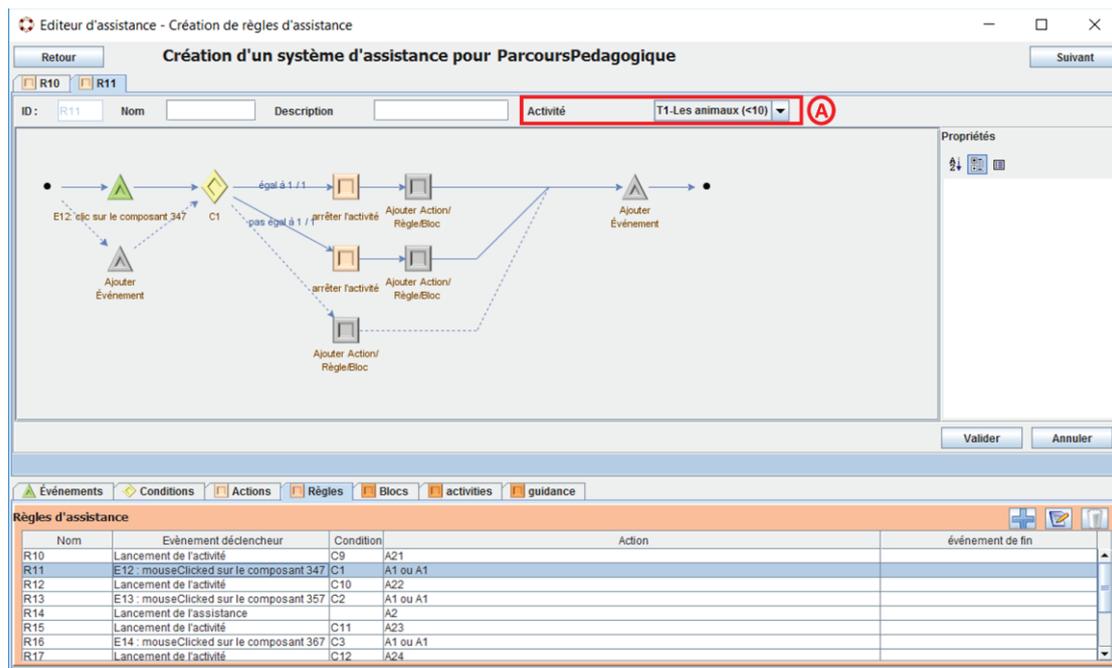


Figure 131 : Définition d'une règle et association de cette règle à l'activité dans SEPIA-edu

5.5.1.4 Description des activités

La description des activités est le résultat de l'étape d'identification des activités. Cette description contient les activités identifiées et leurs états de sortie. Comme certaines règles sont associées aux activités, cette description doit référer à la description des règles.

Pour maintenir la cohérence avec la description des règles, la description des activités est également formalisée en XML. Pour ne pas multiplier les fichiers, la description des activités est stockée dans le même fichier que celui des règles. La Figure 132 représente une partie du fichier qui contient la description des activités. L'association des règles, illustrée dans la Figure 133, aux activités est représentée par l'attribut « activité » dans chaque règle. Par exemple, la règle R10 est associée à l'activité T1.

```

<activites>
  <activite id="T1" nom="A1" commentaire="">
    <etats>
      <etat id="S1" nom="A1_Réussie" commentaire="" idEchelle="1" />
    </etats>
  </activite>
  <activite id="T2" nom="A2" commentaire="">
    <etats>
      <etat id="S2" nom="A2_Réussie" commentaire="" idEchelle="1" />
    </etats>
  </activite>
  <activite id="T3" nom="A3" commentaire="">
    <etats>
      <etat id="S3" nom="A3_Réussie" commentaire="" idEchelle="1" />
    </etats>
  </activite>
</activites>

```

Figure 132 : Description des activités stockée dans le fichier de description des règles aLDEAS de SEPIA-edu

```

<regle id="R7" activite="T1" nomPresentation="" commentaire="" exigePar="0">
  <alternatives condition="C2">
    <alternative valeur="Oui">
      <action id="A1" etats="S1(Réussie)" />
      <action id="A17" />
    </alternative>
  </alternatives>
  <evenement_declencheur idEve="E4" type="mouseClicked" idComp="52" />
</regle>
<regle id="R8" activite="T1" nomPresentation="" commentaire="" exigePar="0">
  <evenement_declencheur idEve="E2" type="lancementActivite" idComp="R8" />
  <alternatives condition="">
    <alternative>
      <action id="A13" />
      <action id="A14" />
    </alternative>
  </alternatives>
  <evenement_de_fin idEve="E4" type="mouseClicked" idComp="52" />
</regle>

```

Figure 133 : Description de l'association des règles aux activités dans le fichier de description des règles aLDEAS de SEPIA-edu

5.5.2 Définition du guidage pédagogique

L'implémentation du patron de guidage pédagogique permet une définition du guidage pédagogique intégrée à l'éditeur d'assistance de SEPIA-edu (cf. Figure 127). Cette étape est dédiée à l'enseignant. Elle lui permet de définir un guidage pédagogique explicitement et assez facilement.

Cette étape est réalisée après l'étape de description de l'interface de l'application-cible, mais avant, après ou pendant l'étape de définition des règles aLDEAS et celle d'identification des activités.

La définition du guidage pédagogique comporte deux sous-étapes principales : la déclaration d'un guidage pédagogique et l'organisation des activités dans le guidage. Le résultat produit dans cette étape est la description du guidage pédagogique transformé en règles aLDEAS.

5.5.2.1 Déclaration d'un guidage pédagogique

Cette étape permet à l'enseignant de déclarer un guidage pédagogique. Pour cela il doit donner les informations générales décrivant le guidage pédagogique, à savoir son nom et sa description. Par exemple, le guidage pédagogique d'identifiant G_3 porte le nom « Comparaison » et sa description est « apprendre à comparer les nombres » (cf. A dans la Figure 134).

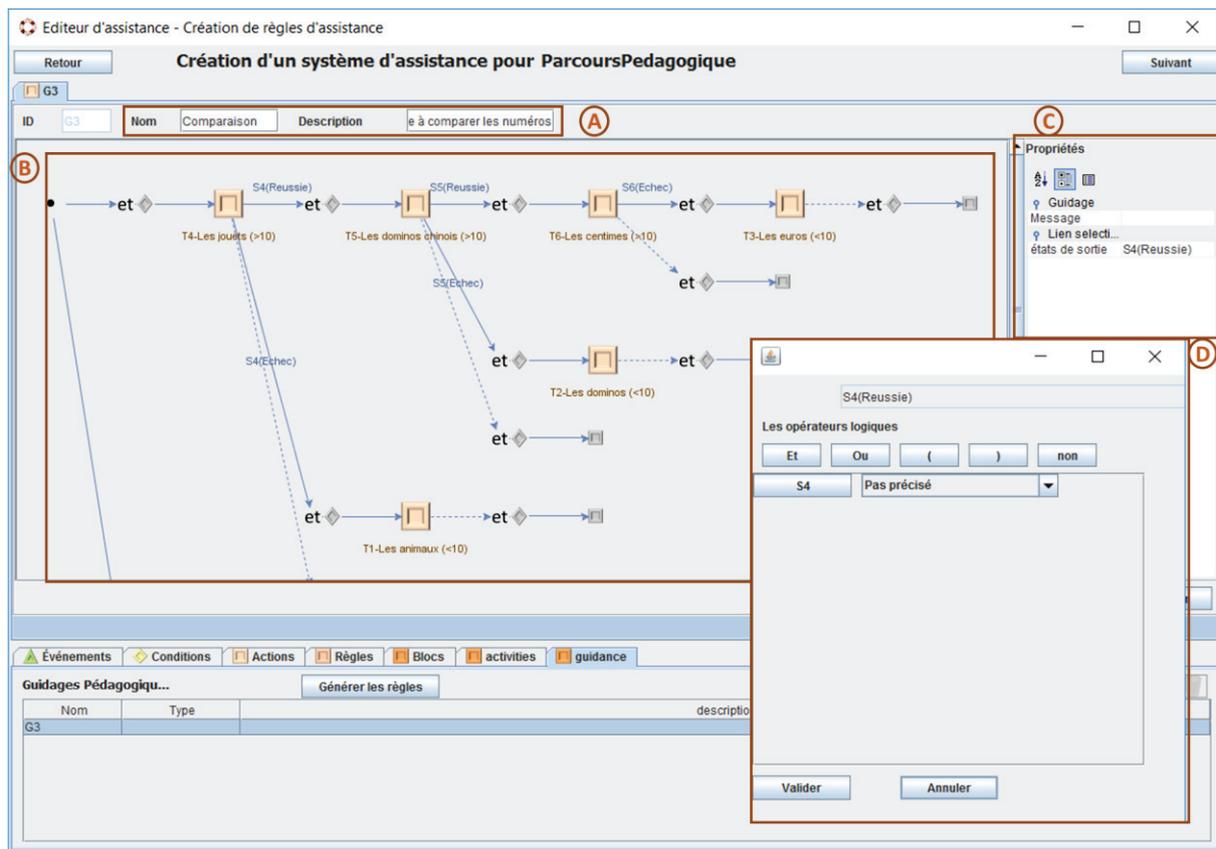


Figure 134 : Écran de définition d'un guidage pédagogique

5.5.2.2 Organisation des activités

Après la déclaration d'un guidage pédagogique, l'enseignant organise les activités dans ce guidage. Pour permettre cette définition, une interface graphique a été proposée (cf. Figure 134) pour implémenter le patron de guidage pédagogique. Cependant, pour simplifier l'affichage du guidage pédagogique, nous ne montrons pas à l'interface la condition « et/ou ». Pour maintenir une définition toujours conforme au patron de guidage pédagogique, la condition « et/ou » reste utilisable, mais implicitement. Ainsi, la condition « et » est utilisée même si l'on ne précise pas de condition « et ». Par contre, pour représenter la condition « ou », il faut dupliquer l'activité-cible. Voyons cela sur un exemple. Nous définissons dans la Figure 135 un guidage pédagogique avec notre patron (cf. partie gauche de la figure) et avec SEPIA-edu (cf. partie droite de la figure). Cet exemple comporte un guidage avec la condition « et » (cf. partie haute de la Figure 135) et un guidage avec la condition « ou » (cf. partie basse de la figure). Pour la condition « et », nous enlevons tout simplement le mot « et ». Pour la condition « ou », nous enlevons la condition « ou » et dupliquons l'activité A₃. Cela permet de proposer une interface plus simple pour les guidages contenant de multiples combinaisons entre activités.

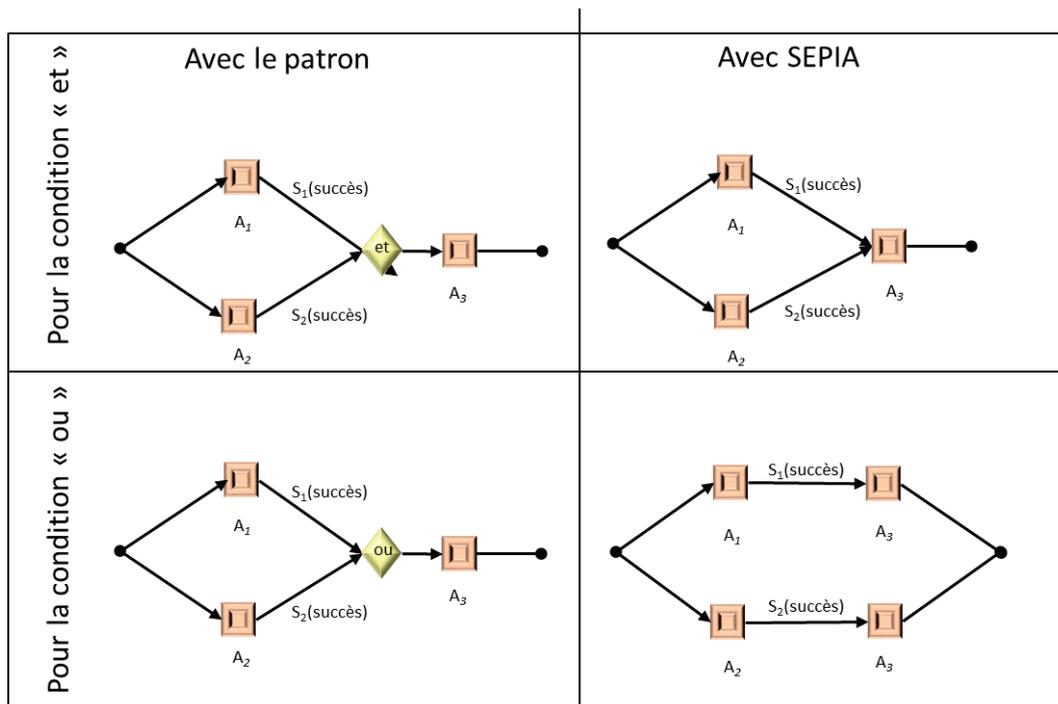


Figure 135 : Illustration de la mise en œuvre de la condition « et/ou » du patron de guidage dans l'éditeur de SEPIA-edu

L'étape d'organisation des activités permet de concevoir un guidage en organisant des activités existantes ou de nouvelles activités à identifier. Grâce à l'interface graphique, le concepteur peut déplacer facilement les activités (cf. B dans la Figure 134). Il peut préciser d'autres informations dans le tableau de propriétés, comme le contenu du message interactif dynamique à afficher aux apprenants pour présenter la liste d'activités (cf. C dans la Figure 134) : ce message est généré par SEPIA-edu, mais il peut contenir du texte précisé par l'enseignant. Par défaut, le contenu du message a été initialisé avec le titre « Liste d'activités » et le contenu textuel « Cliquer sur une activité à faire ». De même, pour la flèche entre les activités, le concepteur peut déterminer les valeurs des états de sortie par le tableau de propriétés (cf. C dans la Figure 134) ou par l'interface textuelle (cf. D dans la Figure 134).

5.5.2.3 Description du guidage pédagogique

La description du guidage pédagogique est le résultat de l'étape de définition du guidage pédagogique. Elle contient le guidage pédagogique et son contenu. Comme un guidage pédagogique contient des activités, cette description doit référer à la description des activités.

Pour maintenir la cohérence avec la description des règles, la description du guidage pédagogique est formalisée en XML. Elle est donc stockée dans un fichier XML. Pour simplifier le stockage de cette description, nous avons utilisé le même fichier que celui de description des règles. La Figure 136 montre un exemple de description d'un guidage pédagogique.

```

<guidages>
  <guidage id="G2" nom="" description="">
    <activite id="T1">
      <contraintes>
        <preActivites intersection="et">
          <activite id="0" />
        </preActivites>
        <condition id="C5" valeur="vrai" />
      </contraintes>
    </activite>
    <activite id="T2">
      <contraintes>
        <preActivites intersection="et">
          <activite id="T1" etats="S1 (Réussie)" />
        </preActivites>
        <condition id="C20" valeur="vrai" />
      </contraintes>
    </activite>
    <activite id="T3">
      <contraintes>
        <preActivites intersection="et">
          <activite id="T2" etats="S2 (Réussie)" />
        </preActivites>
        <condition id="" />
      </contraintes>
    </activite>
    <activite id="T1">
      <contraintes>
        <preActivites intersection="et">
          <activite id="T2" etats="S2 (Échec)" />
        </preActivites>
        <condition id="" />
      </contraintes>
    </activite>
  </guidage>

```

Figure 136 : Description d'un guidage pédagogique stockée dans le fichier de description des règles aLDEAS de SEPIA.

5.5.3 Transformation du guidage pédagogique en règles aLDEAS

Pour que le guidage pédagogique défini soit exécuté par le moteur de SEPIA-edu, il est nécessaire qu'il soit transformé en règles aLDEAS. Pour automatiser cette tâche, nous avons implémenté le processus de transformation du guidage pédagogique en règles aLDEAS.

Cette étape est lancée lors de la sauvegarde par l'enseignant. La transformation donne lieu à la génération et à la modification automatique de règles aLDEAS. Le résultat de cette étape est l'ensemble des règles et conditions générées et/ou modifiées.

Les Figure 137 et Figure 138 illustrent certaines règles et conditions générées automatiquement par l'éditeur d'assistance de SEPIA-edu. Dans cet exemple, les règles RP₆, RP₇, RP₈ et les conditions CP₁, CP₂ et CP₃ sont générés.

```

<regle id="RP2" commentaire="vérification de T1" sceanse="non">
  <alternatives condition="CP1">
    <alternative valeur="vrai">
      <action id="AP1" type="ajouter" idAjoute="T1" nomAjoute="A1" />
    </alternative>
  </alternatives>
  <evenement_declencheur idEve="E1" type="lancementRegle" idComp="RP2" exigePar="0" />
</regle>
<regle id="RP3" commentaire="Choix de T1">
  <alternatives>
    <alternative>
      <action id="R8" />
    </alternative>
  </alternatives>
  <evenement_declencheur type="mouseClicked" idComp="T1" />
</regle>
<regle id="RP4" commentaire="vérification de T2" sceanse="non">
  <alternatives condition="CP2">
    <alternative valeur="vrai">
      <action id="AP1" type="ajouter" idAjoute="T2" nomAjoute="A2" />
    </alternative>
  </alternatives>
  <evenement_declencheur idEve="E1" type="lancementRegle" idComp="RP4" exigePar="0" />
</regle>

```

Figure 137 : Extrait des règles générées par l'éditeur d'assistance de SEPIA-edu

```

<condition id="CP1" type="combinaison" formule="(C5 (vrai) ) || (S2 (Echec) )" />
<condition id="CP2" type="combinaison" formule="(C20 (vrai) ) & & (S1 (Réussie) )" />
<condition id="CP3" type="combinaison" formule="(S2 (Réussie) )" />

```

Figure 138 : Extrait des conditions générées dans l'éditeur d'assistance de SEPIA-edu

5.5.4 Identification des besoins d'assistance

L'identification des besoins d'assistance se base sur le lancement des règles aLDEAS (cf. section 3.5.1.2). Cependant, l'identification des besoins ne tient pas compte du mécanisme d'« activation / désactivation des règles » qui permet d'éviter le lancement de certaines règles concernant des activités autres que celle en cours. Nous avons donc modifié le mécanisme d'« identification des besoins d'assistance » pour que la liste de règles qui attendent des événements contienne uniquement les règles qui n'appartiennent à aucune activité et les règles qui appartiennent à l'activité en cours. Cette liste est mise à jour lorsque les apprenants terminent ou commencent une activité.

5.5.5 Élaboration de la réponse

L'élaboration de la réponse au besoin d'assistance est faite pour exécuter les actions élémentaires d'assistance de la règle. Cependant, cette élaboration ne prenait pas en compte les nouvelles actions élémentaires d'assistance : lancer une activité, arrêter une activité avec des états de sortie et manipuler un message dynamique (ajouter un composant dans le message, afficher le message, vider les composants). Ces actions sont nécessaires pour que le guidage pédagogique puisse être exécuté. Nous les avons donc implémentées dans le moteur de SEPIA-edu.

5.5.6 États de sortie en PMDLe

Le fichier des états de sortie en PMDLe stocke les valeurs des états de sortie des activités. Il est exploité par le moteur d'assistance pour exécuter le guidage pédagogique défini grâce à des règles aLDEAS. Plus précisément, les conditions dans une règle consultent les valeurs des états de sortie. Ces valeurs peuvent être mises à jour par les actions élémentaires d'assistance « arrêter une activité avec des états de sortie ».

5.5.7 Épi-assistants

Les épi-assistants permettent de réaliser les actions élémentaires d'assistance sur les applications-cibles. Cependant, aucun épi-assistant ne permettait de manipuler les messages interactifs dynamiques. Nous avons donc implémenté l'épi-assistant « message interactif dynamique ». Par exemple, la Figure 139 illustre une liste d'activités proposée à l'apprenant par un message interactif dynamique. Ce message permet aux apprenants d'interagir en choisissant une activité à réaliser. Ce choix est envoyé au moteur de SEPIA-edu comme événement de type action de l'utilisateur « clic sur un composant ».

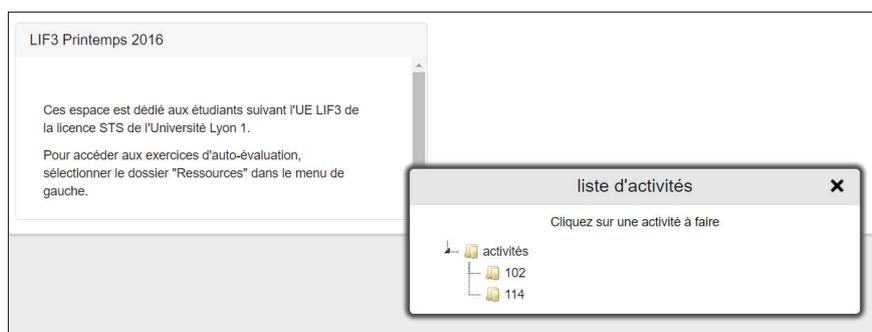


Figure 139 : Exemple du message dynamique qui affiche une liste d'activités dans SEPIA-edu

5.6. Exemple d'un guidage pédagogique défini avec SEPIA-edu

Dans cette section, nous présentons un exemple concret de guidage pédagogique défini avec l'éditeur SEPIA-edu. Le système d'assistance est donc constitué d'un ensemble de règles décrivant ce guidage pédagogique composé d'activités organisées.

Notre exemple est celui d'un système d'assistance pour le MOOC Fun « Racines » qui donne des bases d'étymologie à de futurs étudiants de sciences et de santé. Il propose sur plusieurs semaines des cours, des exercices et un certain nombre d'outils classiques pour un MOOC (forum, FAQ, index, etc.). Le système d'assistance que nous avons défini ajoute un guidage pédagogique sur ce MOOC pour qu'il soit en adéquation avec les objectifs donnés d'un enseignant donné pour un contexte donné. Ce guidage pédagogique intègre trois activités pour

le jour 3 de la semaine 1 du MOOC « Racines ». L'exécution de ce système d'assistance est illustrée par une vidéo en ligne sur le site du projet AGATE¹ (cf. Figure 140).



Figure 140 : Vue d'ensemble du système d'assistance qui propose à l'apprenant d'une part une liste d'activités organisées dans un guidage pédagogique et d'autre part de l'assistance à la réalisation d'une activité sélectionnée

Nous allons à présent détailler le contenu de ce système d'assistance en montrant comment il est mis en œuvre en respectant le langage aLDEAS à l'aide des modèles et outils que nous avons définis concernant le guidage pédagogique, les activités et règles définis. Avant de lancer l'assistance, l'apprenant est déjà sur la page du jour 3 du MOOC. Nous décrivons ici les trois activités qui composent ce guidage pédagogique en instant sur l'enchaînement entre ces activités.

- Dans l'activité A_1 , l'apprenant doit visionner la vidéo du jour 3 « Racines Blast(o)- et Germ- » (cf. Figure 141). Le système d'assistance guide l'apprenant vers « Racines Blast(o)- et Germ- » dans le menu dédié aux activités du jour 3. Le système d'assistance lui demande de regarder la vidéo du cours. Lorsque l'apprenant a terminé de visionner la vidéo, il l'indique au système en répondant à une consultation de l'utilisateur, le système le alors dirige vers l'activité A_2 .

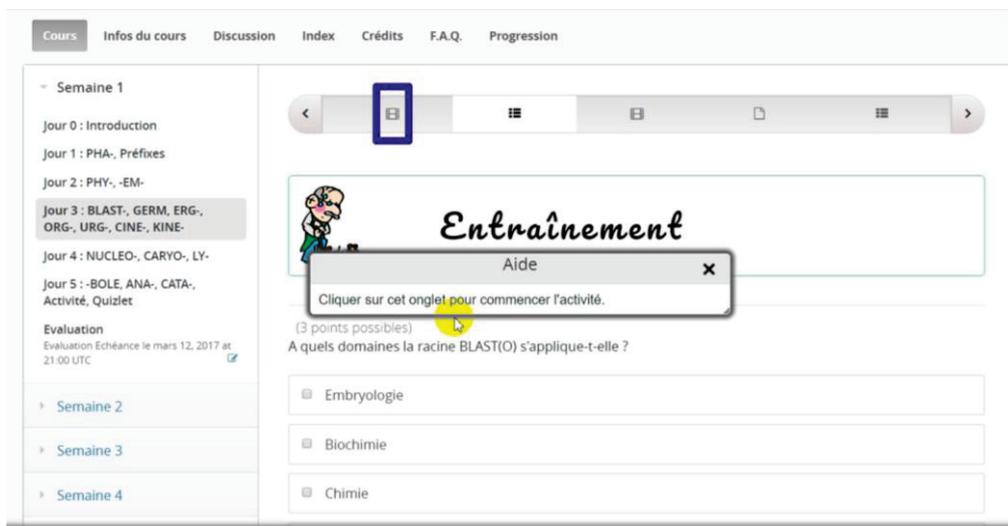


Figure 141 : Exécution de l'étape « guider l'apprenant vers l'activité » de l'activité A_1 du système d'assistance du MOOC « Racines » avec la version Web du moteur SEPIA-edu

En aLDEAS :

Activité : id = T_1 , nom = A_1 (cf. Figure 142)

¹ Lien de la vidéo de l'exécution du système d'assistance du MOOC Fun « Racines » : <https://www.youtube.com/watch?v=i-ajnnft2E4>

États de sortie : Passage de A₁ (type= « textuel », valeurs = « oui »)

Règles aLDEAS associées : R₇ (cf. Figure 143), R₈ (cf. Figure 144)

R₈ permet de guider l'apprenant vers l'onglet « Racines Blast(o)- et Germ- ». R₈ débute par le lancement de l'activité (A₁). R₈ effectue les actions A₁₃ et A₁₄.

R₇ permet de demander à l'apprenant de regarder la vidéo et de savoir quand l'apprenant termine le visionnage de la vidéo pour mettre fin à l'activité A₁ en complétant ses états de sortie. R₇ débute par E₄. Comme nous n'avons pas avec SEPIA de moyen automatique de déterminer le moment où l'apprenant termine de visionner la vidéo, R₇ pose directement la question à l'apprenant *via* une consultation utilisateur C₂. Le cas échéant, R₇ effectue l'action « arrêter l'activité » pour mettre fin à l'activité en complétant ses états de sortie « Passage de l'activité A₁ (oui) », puis effectue A₁₇.

E₄ : type d'événement = « action de l'utilisateur », action = « clic », composant = « 52 » (onglet « Racines Blast(o)- et Germ - »)

C₂ : type de consultation = « consultation utilisateur », texte = « Regardez la vidéo ! C'est fait ? », choix = {« C'est fait »}

A₁₃ : type d'action = « message », texte = « Cliquer sur cet onglet pour commencer l'activité. »

A₁₄ : type d'action = « mise en valeur », type de mise en valeur = « entourer », composant = « 52 » (onglet « Racines Blast(o)- et Germ - »)

A₁₇ : type d'action = message, texte = « L'activité A₁ est terminée. »

The screenshot shows the SEPIA-edu editor interface for activity A1. The window title is 'T1'. The interface is divided into two main sections: 'Les règles associées à cette activité' and 'Liste des états de sortie'.

Les règles associées à cette activité

Nom	Evènement déclencheur	Condition	Action	évènement de fin
R7	E4 : mouseClicked sur le composant 52	C2	A1, A17	
R8	Lancement de l'activité		A13, A14	E4 : clic sur le composant 52

Liste des états de sortie

ID	Nom	Commentaire	Echelle
S1	A1_Passage		Passage

Figure 142 : Identification de l'activité A₁ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

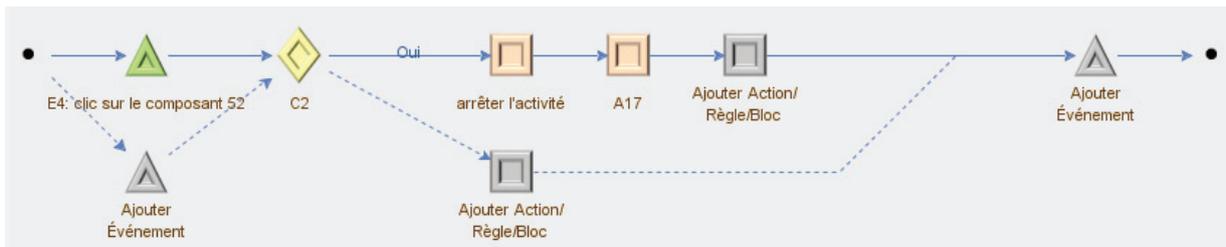


Figure 143 : Définition de R_7 avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

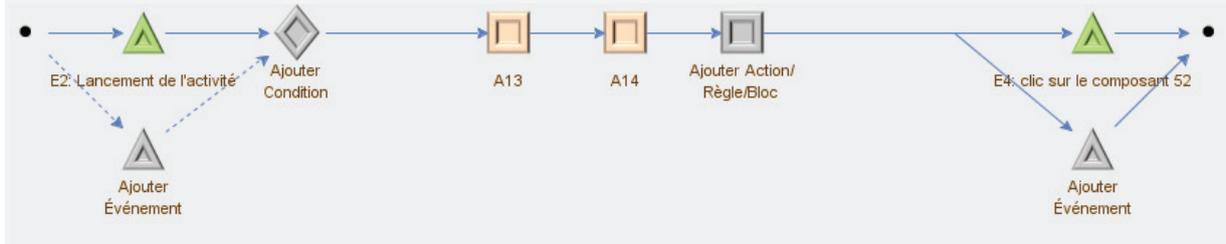


Figure 144 : Définition de R_8 avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

- Dans l'activité A_2 , l'apprenant doit s'entraîner avec le questionnaire. Le système d'assistance guide l'apprenant vers l'onglet « entraînement » (cf. Figure 145). Lors du clic sur le bouton « vérifier la réponse », le système d'assistance vérifie les réponses de l'apprenant. Si l'apprenant réussit les exercices (c'est-à-dire s'il donne les bonnes réponses au questionnaire), le système le dirige vers l'activité A_3 . Sinon, le système lui demande de refaire l'activité A_1 . Pour simplifier, le système d'assistance de notre exemple ne prend en compte que la réponse à la deuxième question.

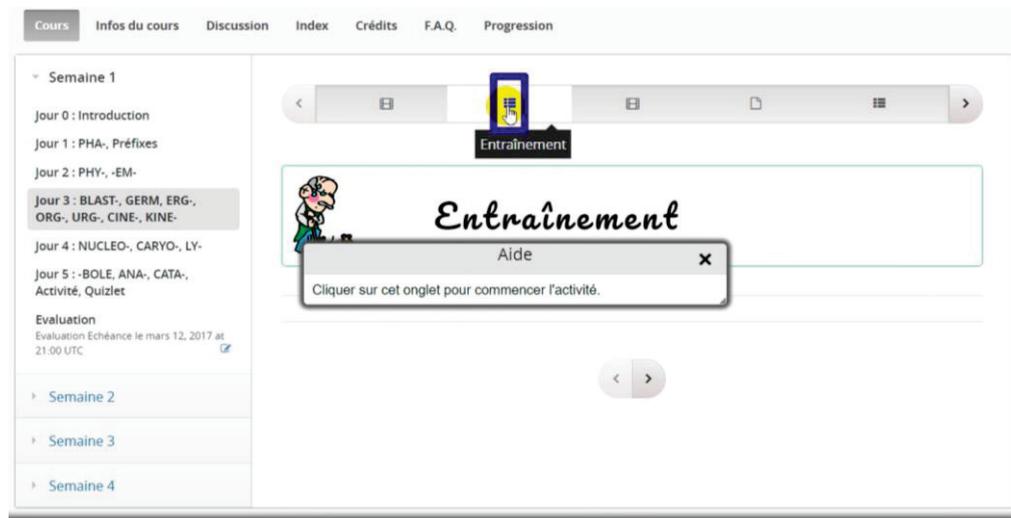


Figure 145 : Exécution de l'étape « guider l'apprenant à l'activité » de l'activité A_2 du système d'assistance du MOOC « Racines » avec la version Web du moteur SEPIA-edu

En aLDEAS :

Activité : id = T_2 , nom = A_2 (cf. Figure 146)

États de sortie : Réussie de l'activité A₂ (type= « textuel », valeurs = « Réussie / Échec »}

Règles aLDEAS associées : R₁₁ (cf. Figure 147), R₁₂ (cf. Figure 148)

R₁₁ permet de guider l'apprenant à l'onglet « entraînement ». R₁₁ débute par le lancement de l'activité (A₂). R₁₁ effectue l'action A₁₃, A₁₈, A₂₀.

R₁₂ permet de vérifier la réponse de l'apprenant et de mettre fin à l'activité en complétant les états de sortie. R₇ débute par E₆ (clic sur le bouton « vérifier la réponse »). R₁₂ vérifie la réponse *via* une combinaison de consultations de l'état de l'application C₁₈.

- Si cette combinaison renvoie la valeur « vrai » (la réponse est correcte), le système effectue l'action « arrêter l'activité » pour mettre fin à l'activité en complétant l'état de sortie avec « Réussie de l'activité A₂ (Réussie) » et effectue A₂₁.
- Sinon, le système met fin à l'activité avec l'état de sortie « Réussie de l'activité A₂ (Échec) » et effectue A₂₅.

E₆ : type d'événement = « action de l'utilisateur », action = « clic », composant = « 115 » (onglet « entraînement »)

E₇ : type d'événement = « action de l'utilisateur », action = « clic », composant = « 53 » (bouton « vérifier la réponse »)

C₁₇ : type de consultation = « consultation de l'état de l'application », composant = « 6 » (réponse de l'utilisateur à la deuxième question), propriété = « texte »

C₁₈ : type de consultation = « combinaison », formule = « C17 (égal à "hématoblaste") || C17 (égal à "hématoblastes") »

A₁₈ : type d'action = « mise en valeur », type de mise en valeur = « entourer », composant = « 115 » (entraînement)

A₂₁ : type d'action = message, texte = « L'activité A₂ est terminée avec succès. »

A₂₅ : type d'action = message, texte = « L'activité A₂ est terminée avec échec. Je te conseille à refaire l'activité A₁. »

T1 T2 T3

ID T2 Nom A2 : Entraînement Description appartenant à l'activité

Les règles associées à cette activité

Nom	Evènement déclencheur	Condition	Action	évènement de fin
R11	Lancement de l'activité		A13, A18, A20	E7 : clic sur le composant 53
R12	E6 : mouseClicked sur le composant 115	C18	A1, A21 ou A1, A25	

Liste des états de sortie

ID	Nom	Commentaire	Echelle
S2	A2_Réussie		Réussie/Échec

Figure 146 : Identification de l'activité A₂ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

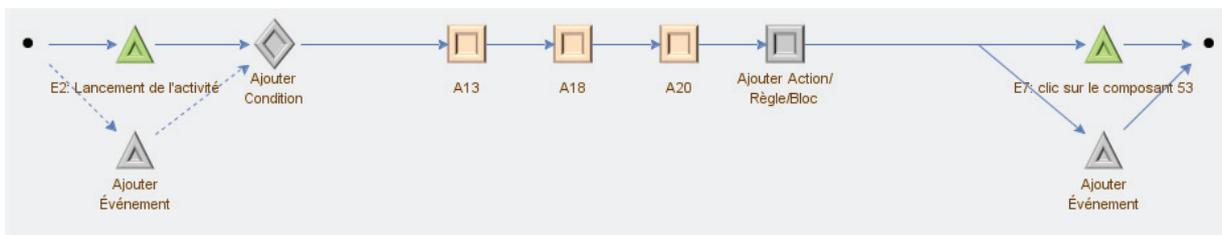


Figure 147 : Définition de la règle R₁₁ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

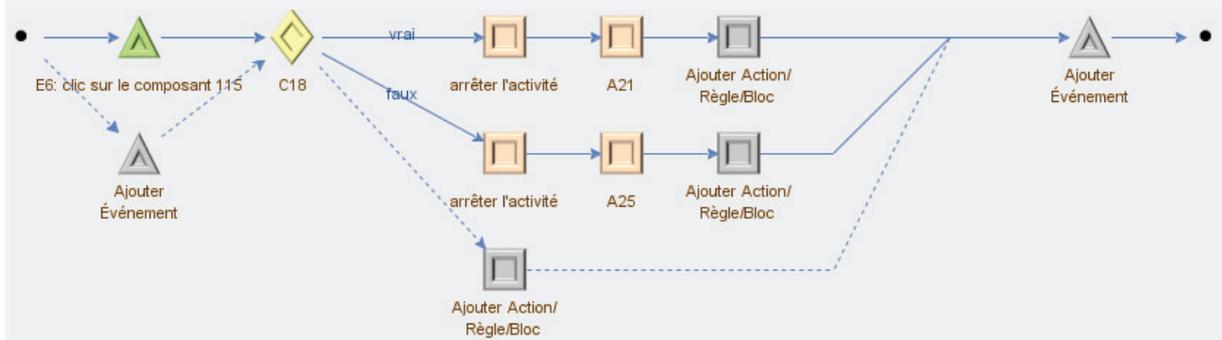


Figure 148 : Définition de la règle R₁₂ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

- Dans l'activité A3, l'apprenant doit regarder la vidéo du jour 3 « Racine Erg(o)-, Org-, Urg- ». Le système d'assistance guide l'apprenant vers cet onglet (cf. Figure 149). Il lui demande de visionner la vidéo. À la fin de la vidéo, l'activité est également terminée.

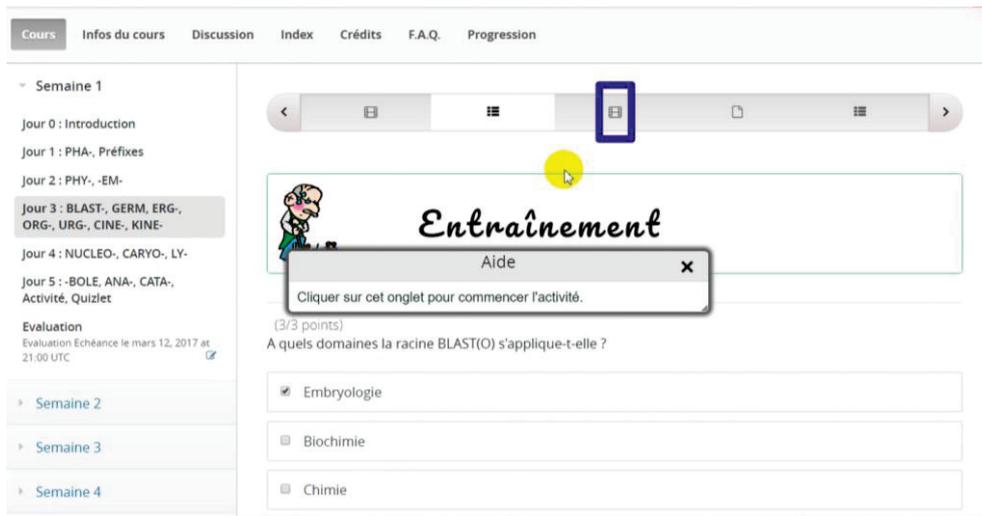


Figure 149 : Exécution de l'étape « guider l'apprenant à l'activité » de l'activité A₂ du système d'assistance du MOOC « Racines » avec la version Web du moteur SEPIA-edu

En alDEAS :

Activité : id = T₃, nom = A₃ (cf. Figure 150)

États de sortie : Passage de A₁ (type= « textuel », valeurs = « oui »)

Règles associées : R₁₃ (cf. Figure 151), R₈ (cf. Figure 152)

R₁₃ permet de guider l'apprenant vers l'onglet « Racine Erg(o)-, Org-, Urg- ». R₁₃ débute par le lancement de l'activité (A₃). R₁₃ effectue l'action A₁₃ et A₁₉.

R₁₄ permet de demander à l'apprenant de regarder la vidéo, de savoir quand l'apprenant termine la vidéo pour mettre fin à l'activité en complétant ses états de sortie. R₁₄ débute par E₈. Comme nous n'avons pas de moyen avec SEPIA de déterminer le moment où l'apprenant termine la vidéo, R₁₄ demande à l'apprenant s'il a terminé la vidéo *via* une consultation utilisateur C₂. Si c'est fait, R₇ effectue l'action « arrêter l'activité » pour mettre fin à l'activité avec l'état de sortie « Passage de l'activité A₃ (oui) » et effectue A₂₄.

E₈ : type d'événement = « action de l'utilisateur », action = « clic », composant = « 54 » (onglet « Racine Erg(o)-, Org-, Urg- »)

C₂ : type de consultation = « consultation utilisateur », texte = « Regardez la vidéo ! C'est fait ? », choix = {« C'est fait »}

A₁₃ : type d'action = « message », texte = « Cliquer sur cet onglet pour commencer l'activité. »

A₁₉ : type d'action = « mise en valeur », type de mise en valeur = « entourer », composant = « 54 » (« Racine Erg(o)-, Org-, Urg- »)

A₂₄ : type d'action = « message », texte = « L'activité A₃ est terminée. »

T1 T2 T3

ID T3 Nom Erg(o)-, Org-, Urg- Description appartenant à l'activité

Les règles associées à cette activité

Nom	Evènement déclencheur	Condition	Action	évènement de fin
R13	Lancement de l'activité		A13, A19	E8 : clic sur le composant 54
R14	E8 : mouseClicked sur le composant 54	C2	A1, A24	

Liste des états de sortie

ID	Nom	Commentaire	Echelle
S3	A3_Passage		Passage

Figure 150 : Identification de l'activité A₃ dans l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

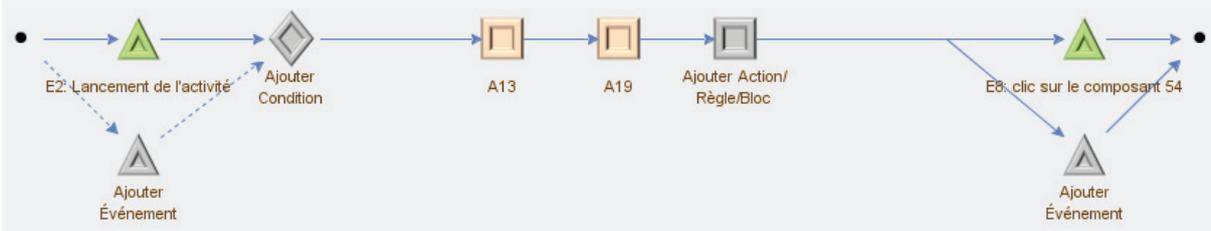


Figure 151: Définition de la règle R₁₃ dans l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

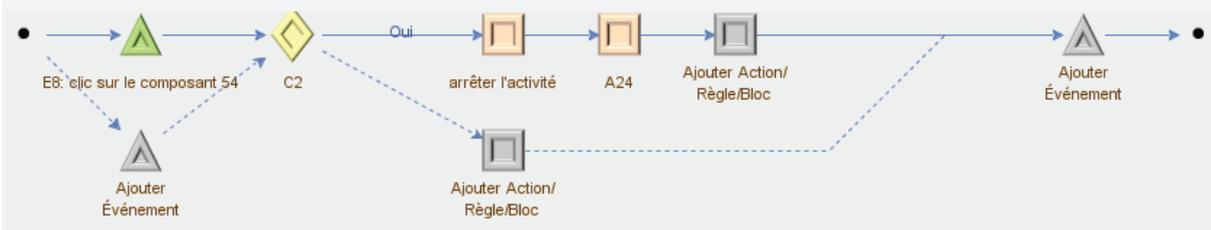


Figure 152 : Définition de la règle R₁₄ avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

Ces activités A₁, A₂, A₃ sont organisées dans un guidage pédagogique. Lors du lancement de l'assistance, le système lance le guidage pédagogique (cf. Figure 153).

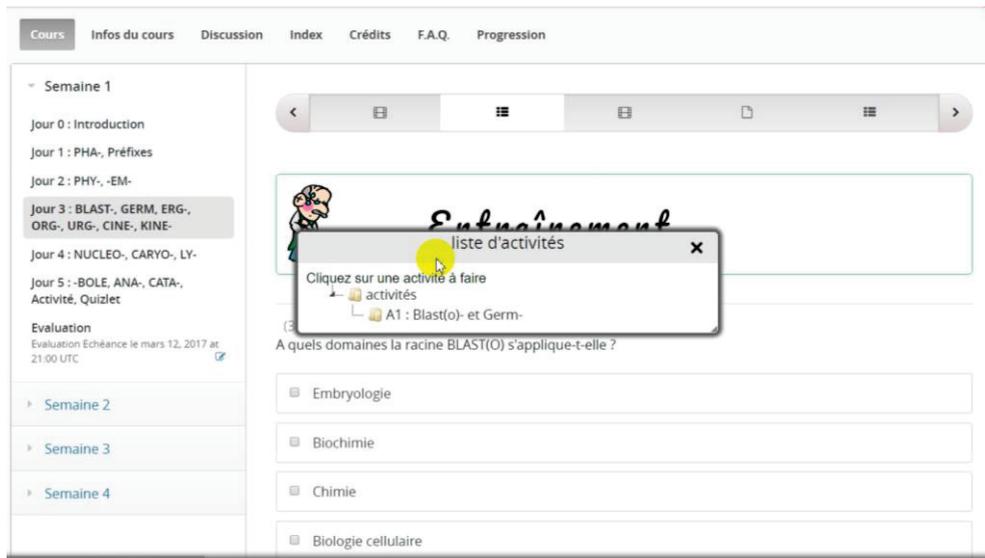


Figure 153 : Exécution du guidage pédagogique « Affichage une liste d'activités » du système d'assistance du MOOC « Racines » avec la version Web du moteur SEPIA-edu

En aLDEAS :

Guidage pédagogique : id = G₁, nom = G₁

Activités organisées : A₁, A₂, A₃

Éléments générés/ modifiés : 10 règles générées, 1 règle modifiée, 3 conditions générées (cf. Figure 155)

Le guidage débute par le lancement de l'assistance. Il propose tout d'abord à l'apprenant l'activité A₁. Une consultation de l'état de sortie C₅ avant l'activité A₁ permet de ne pas proposer cette activité si l'apprenant passe déjà cette activité.

Lors de la terminaison de cette activité, le guidage propose l'activité A₂. Comme pour l'activité A₁, la combinaison de consultations C₂₀ avant l'activité A₂ permet de ne pas proposer A₂ si l'apprenant a déjà réussi cette activité.

- Si l'apprenant réussit cette activité, le guidage propose l'activité A₃.
- Sinon, le guidage propose l'activité A₁ comme activité de remédiation. Dans ce cas, le guidage affiche une liste d'activités avec deux activités : A₁ et A₂.

C₅ : type de consultation= « consultation de l'état de sortie », composant = « S1 » (passage de l'activité A₁)

C₁₉ : type de consultation= « consultation de l'état de sortie », composant = « S2 » (réussite de l'activité A₂)

C₂₀ : type de consultation = « combinaison », formule = « !C₁₉(Réussie) »

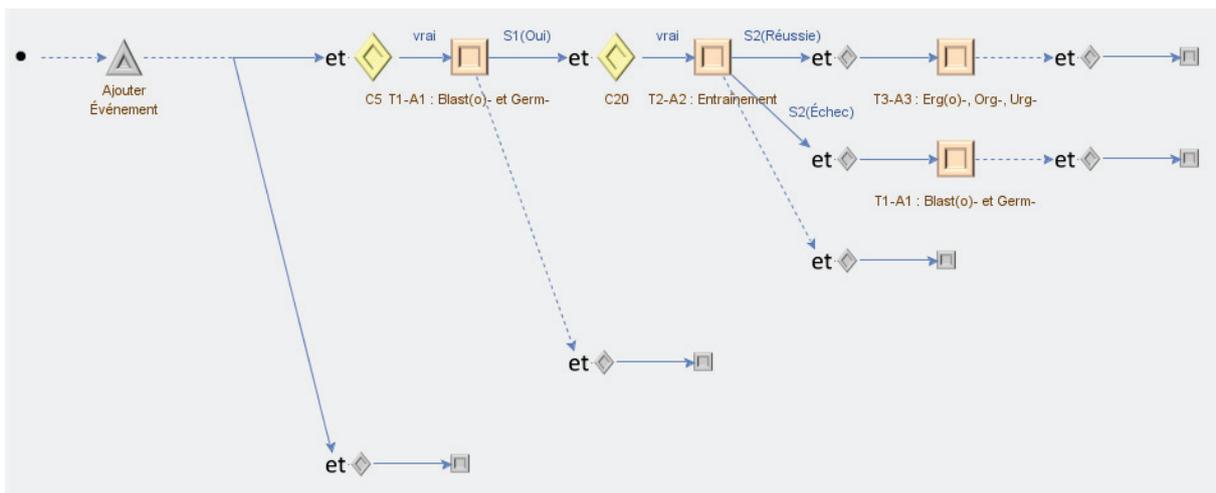


Figure 154 : Définition du guidage pédagogique G_1 pour le système d'assistance du MOOC « Racines » avec l'éditeur SEPIA-edu

Transformation du guidage pédagogique : lors de l'enregistrement du guidage pédagogique, les éléments aLDEAS sont automatiquement modifiés et générés : 10 règles sont générées, 1 règle est modifiée, 3 conditions sont générées (cf. Figure 155).

```

<regle id="RP2" commentaire="vérification de T1" sceance="non">
  <alternatives condition="CP1">
    <alternative valeur="vrai">
      <action id="AF1" type="ajouter" idAjoute="T1" nomAjoute="A1 : Blast(o)- et Germ-" />
    </alternative>
  </alternatives>
  <evenement_declencheur idEve="E1" type="lancementRegle" idComp="RP2" exigePar="0" />
</regle>
<regle id="RP3" commentaire="Choix de T1">
  <alternatives>
    <alternative>
      <action id="R8" />
    </alternative>
  </alternatives>
  <evenement_declencheur type="mouseClicked" idComp="T1" />
</regle>
<regle id="RP4" commentaire="vérification de T2" sceance="non">
  <alternatives condition="CP2">
    <alternative valeur="vrai">
      <action id="AF1" type="ajouter" idAjoute="T2" nomAjoute="A2 : Entraînement" />
    </alternative>
  </alternatives>
  <evenement_declencheur idEve="E1" type="lancementRegle" idComp="RP4" exigePar="0" />
</regle>
<regle id="RP5" commentaire="Choix de T2">
  <alternatives>
    <alternative>
      <action id="R11" />
    </alternative>
  </alternatives>
  <evenement_declencheur type="mouseClicked" idComp="T2" />
</regle>

```

Figure 155 : Génération de règles avec l'éditeur SEPIA-edu pour le système d'assistance du MOOC « Racines »

Dans ce chapitre, nous avons proposé (1) un **modèle d'activité** qui permet d'identifier les activités sur les applications-cibles, (2) un **modèle de guidage pédagogique** explicitant les cinq types de guidages pédagogiques : libre, séquentiel, contextuel, personnalisé et temporel, (3) un **patron de guidage pédagogique** qui permet de définir un guidage pédagogique complet combinant différents types et (4) un **processus de transformation d'un guidage en règles aLDEAS** qui permet de générer et de modifier des règles aLDEAS. Le modèle d'activité, le patron de

guidage pédagogique et le processus de transformation ont été implémentés dans SEPIA-edu en y ajoutant deux étapes destinées aux enseignants de « définition des activités » et de « définition du guidage pédagogique », ainsi qu'une étape automatisée de « transformation du guidage pédagogique en règles aLDEAS ». Les enseignants peuvent donc créer des activités et créer un guidage pédagogique grâce à l'interface graphique.

Chapitre 6. Évaluation

Objectif de chapitre

Évaluer notre approche et nos contributions.

Introduction de ce chapitre

L'objectif de cette thèse est de permettre la mise en place d'un système d'assistance en contexte éducatif. Pour rendre la définition de l'assistance plus accessible aux ingénieurs pédagogiques, nous avons proposé deux contributions présentées dans les deux chapitres précédents. Ces contributions rendre plus explicite et plus facile la définition de l'articulation entre les règles du langage aLDEAS ainsi que la définition d'un guidage pédagogique avec le système SEPIA.

Ce chapitre du manuscrit concerne l'évaluation de nos contributions théoriques et de leur mise en œuvre. Nous présentons tout d'abord dans la section 6.1 une grille d'évaluation présentant sous forme synthétique les différents aspects à évaluer pour valider nos propositions. Dans la section 6.2, nous présentons de manière détaillée la réalisation des mises à l'essai que nous avons conduites.

6.1. Grille d'évaluation

Nous présentons dans cette section les différentes facettes à évaluer en précisant les questions que cette évaluation soulève. Pour évaluer chaque facette, nous précisons les méthodes d'évaluation que nous avons employées, ainsi que les critères d'évaluation que nous avons définis. Pour chaque critère d'évaluation, nous indiquons le résultat des évaluations que nous avons effectuées sont trois niveaux : validé ✓, partiellement validé ✓ et non validé ✗. Dans la section suivante, nous présentons les mises à l'essai que nous avons mises en place pour évaluer ces différentes facettes.

6.1.1 Évaluation des modèles du guidage pédagogique et de l'articulation entre règles

Le principe des modèles proposés dans cette thèse est d'exprimer les éléments à un niveau supérieur par rapport au niveau des règles aLDEAS : le niveau de l'articulation entre règles et le niveau du guidage pédagogique, ou même le niveau des activités qui constituent un guidage pédagogique. Pour évaluer ces modèles et le patron associé, nous devons établir leur capacité d'expression pour définir l'articulation entre règles, le guidage pédagogique et les activités

Tout d'abord, la première question concerne les concepteurs d'assistance ciblés, principalement des enseignants et les ingénieurs pédagogiques : « les modèles sont-ils compréhensibles par les concepteurs ? ». La deuxième question concerne également les concepteurs d'assistance : « les modèles permettent-ils d'exprimer explicitement le déroulement d'une assistance et le guidage pédagogique ». La troisième question concerne la couverture du modèle d'articulation entre règles et du patron de guidage pédagogique : « les modèles permettent-ils d'exprimer des déroulements d'assistance et des guidages pédagogiques variés ? ». La dernière question concerne la mise en œuvre de nos modèles : « les modèles sont-ils implémentables ? ».

Facette à évaluer	Méthode d'évaluation	Critère d'évaluation	Résultat	Détails
Les modèles sont-ils compréhensibles par les concepteurs ?	Questionnaire auprès de concepteurs d'assistance	Les concepteurs comprennent les modes d'articulation	✓	6.2.4
		Les concepteurs comprennent les guidages pédagogiques	✗	6.2.5
	Mise à l'essai auprès de concepteurs d'assistance	Les concepteurs reconnaissent les modes d'articulation entre règles	✓	6.2.4
		Les concepteurs reconnaissent les guidages pédagogiques	✗	6.2.5
Les modèles permettent-ils	Mise à l'essai auprès de concepteurs d'assistance	Les concepteurs réussissent à créer, compléter, modifier	✓	6.2.4

d'exprimer explicitement le déroulement d'une assistance et le guidage pédagogique		une assistance en utilisant le modèle d'articulation entre règles		
		Les concepteurs réussissent à créer, compléter, modifier une assistance en utilisant le modèle d'activité et le patron de guidage pédagogique	✗	6.2.5
Les modèles permettent-ils d'exprimer des déroulements d'assistance et des guidages pédagogiques variés ?	Utilisation du modèle d'articulation pour définir le déroulement d'assistances existantes	Le modèle d'articulation permet de définir des assistances existantes variées	✓	6.2.2
	Mise à l'essai avec des concepteurs d'assistance pour définir des systèmes d'assistance en utilisant les modes d'articulation entre règles	L'utilisation des modes d'articulation entre les règles permet à des concepteurs d'assistance de définir l'assistance qu'ils souhaitent	✓	6.2.4
	Utilisation du modèle de guidage pour définir des guidages pédagogiques existants	Le modèle permet de définir des guidages existants variés	✓	6.2.2
	Mise à l'essai avec des concepteurs d'assistance pour définir des systèmes d'assistance en utilisant les modes d'articulation entre règles	L'utilisation des modes d'articulation entre règles permet à des concepteurs d'assistance de définir l'assistance qu'ils souhaitent	✗	6.2.5
Les modèles sont-ils implémentables ?	Implémentation des modèles	L'implémentation des modèles dans SEPIA-edu permet la définition d'un système d'assistance	✓	6.2.2

Tableau 3 : Évaluation des modèles

Le Tableau 3 montre que les modèles proposés pour l'articulation entre règles sont validés par leur utilisation sur certaines applications et par l'implémentation dans SEPIA-edu. Cependant, les modèles proposés pour le guidage pédagogique ne sont pas validés par la première mise à l'essai menée (cf. section 6.2.5), bien que nous les ayons utilisés par ailleurs avec succès pour définir des guidages pédagogiques. De plus, pour nos mises à l'essai, nous n'avons pas pu travailler avec notre public cible, que sont les enseignants et les ingénieurs pédagogiques. Nous avons travaillé avec des étudiants de l'Université Lyon 1 qui ont joué le rôle de concepteurs d'assistance.

6.1.2 Évaluation des processus de transformation d'un guidage pédagogique en règles aLDEAS et d'application automatique des contraintes aux règles aLDEAS

Si nos modèles permettent d'exprimer l'articulation entre règles aLDEAS et d'exprimer le guidage pédagogique, le processus de transformation d'un guidage pédagogique en règles aLDEAS (intégrant le processus d'application automatique des contraintes aux règles et le processus de transformation du guidage pédagogique en règles aLDEAS) doit permettre de transformer l'articulation et le guidage en règles aLDEAS. Pour évaluer ces processus, nous devons évaluer dans quelle mesure ils facilitent la définition d'un système d'assistance.

Tout d'abord, il est nécessaire d'évaluer la possibilité d'implémentation des processus : « les processus sont-ils implémentables ? ». Ensuite, nous devons nous assurer que la transformation est correcte : « les règles générées assurent-elles une exécution correcte de l'assistance ? ». Puis nous devons vérifier si nos propositions permettent de limiter les erreurs que peuvent rencontrer les concepteurs d'assistance : « les processus limitent-ils les erreurs des concepteurs ? ». Enfin, nous étudions la quantité de tâches que doivent réaliser les enseignants pour définir une assistance : « les processus permettent-ils de réduire les tâches ».

Facette à évaluer	Méthode d'évaluation	Critère d'évaluation	Résultat	Détails
Les processus sont-ils implémentables ?	Implémentation des processus	Application automatique des contraintes sur les règles	✓	6.2.2
Les règles générées assurent-elles une exécution correcte de l'assistance ?	Exécution de règles générées	Les règles converties sont compatibles avec le moteur d'assistance de SEPIA	✓	6.2.2
Les processus limitent-ils les erreurs des concepteurs ?	Évaluation analytique	Liste d'erreurs possibles évitées pour les concepteurs	✓	6.2.2
Les processus permettent-ils de réduire les tâches ?	Évaluation analytique	Les processus permettent de réduire le nombre des tâches	✓	6.2.2

Tableau 4 : Évaluation des processus de transformation en règles aLDEAS

Le Tableau 4 montre que toutes les facettes sont validées. Cela montre que l'implémentation de nos processus peut automatiser certaines tâches des concepteurs en réduisant les possibilités d'erreurs.

6.1.3 Évaluation du système SEPIA-edu

Le système SEPIA a été enrichi grâce à l'implémentation de nos modèles et de nos processus. L'évaluation de nos modèles et processus doit donc être complétée par l'évaluation de leur mise en œuvre dans SEPIA-edu.

Tout d'abord, il est nécessaire d'évaluer la compréhension par un concepteur d'assistance d'un système d'assistance défini avec SEPIA-edu : « est-ce qu'un concepteur d'assistance peut comprendre une assistance de SEPIA-edu ? ». En second lieu, il convient d'étudier la définition d'un système d'assistance avec SEPIA par le concepteur : « est-ce qu'un concepteur d'assistance peut utiliser SEPIA-edu pour définir une assistance ? ». L'évaluation concerne également les bénéfices apportés par SEPIA-edu par rapport à la version initiale de SEPIA (facilitation de l'utilisation, intuitivité, réduction des tâches, du temps) : « est-ce que l'utilisation de SEPIA-edu plutôt que SEPIA est bénéfique ? ». L'évaluation doit également s'intéresser à l'ergonomie de SEPIA-edu : « SEPIA-edu est-il ergonomique ? ». Enfin, nous devons étudier la couverture de SEPIA-edu : « Est-ce que l'utilisation de SEPIA-edu répond aux différentes attentes des concepteurs ? ».

Facette à évaluer	Méthode d'évaluation	Critère d'évaluation	Résultat	Détails
Est-ce qu'un concepteur d'assistance peut comprendre une assistance de SEPIA-edu ?	Mise à l'essai auprès de concepteurs	Ils peuvent comprendre le déroulement d'une assistance dans SEPIA-edu	✓	6.2.4
		Ils peuvent comprendre le guidage pédagogique de SEPIA-edu	✗	6.2.5
Est-ce qu'un concepteur d'assistance peut utiliser SEPIA-edu pour définir une assistance ?	Mise à l'essai auprès de concepteurs	Ils peuvent définir le déroulement d'une assistance dans SEPIA-edu	✓	6.2.4
		Ils peuvent définir un guidage pédagogique dans SEPIA-edu	✗	6.2.5
Est-ce que l'utilisation de SEPIA-edu plutôt que SEPIA est bénéfique ?	Mise à l'essai auprès de concepteurs	En utilisant SEPIA-edu pour définir le déroulement d'une assistance, ils peuvent gagner : - Facilité d'utilisation - Intuitivité - Réduction d'erreurs, temps	✓	6.2.4
		En utilisant SEPIA-edu pour définir le guidage pédagogique, ils peuvent gagner : - Facilité d'utilisation - Intuitivité - Réduction d'erreurs, temps	✗	6.2.5
SEPIA-edu est-il ergonomique ?	Évaluation ergonomique	Les critères classiques d'évaluation de l'IHM sont respectés	✓	
Est-ce que l'utilisation de SEPIA-edu répond aux différentes attentes des concepteurs ?	Questionnaire auprès de concepteurs	Les modèles permettent de mettre en place le déroulement des assistances souhaitées	✓	6.2.4

		Les modèles permettent de mettre en place le guidage pédagogique souhaité	X	6.2.5
--	--	---	---	-------

Tableau 5 : Évaluation de SEPIA-edu

Le Tableau 5 montre que le système SEPIA-edu est validé du point de vue de l'utilisation des blocs pour la définition d'un système d'assistance. Cependant, la mise à l'essai de 2017 n'a pas validé la définition d'un guidage pédagogique avec SEPIA-edu. Cela nous impose de futures améliorations du système pour faire des expérimentations.

6.2. Évaluations réalisées

Dans la section précédente, notre grille d'évaluation nous a permis d'identifier les facettes à évaluer, les méthodes d'évaluation et les critères d'évaluation permettant de valider notre travail de recherche. Dans cette section, nous présentons les évaluations que nous avons conduites dans l'objectif de cette validation. Les évaluations menées le sont de différentes manières : par des études sur l'assistance (6.2.1), par l'analyse des processus de transformation en règles aLDEAS (6.2.2), par l'implémentation des modèles et des processus (6.2.2), ainsi que par des mises à l'essai (6.2.4. et 6.2.5).

6.2.1 Études sur l'assistance

Depuis les travaux classiques de (Vygotsky, 1962) et de (Bruner, 1985) notamment, l'assistance dans le contexte éducatif est considérée comme un élément central pour construire des savoirs, savoir-faire et savoir-être. Ceci est vrai en particulier sur les applications informatiques.

L'importance du rôle de l'assistance dans l'apprentissage a pour conséquence des besoins importants de mise en place d'assistances destinées aux apprenants, par les cadres de la situation d'apprentissage, parmi lesquels les enseignants. Ainsi, d'une part des travaux ont étudié l'interaction entre les tuteurs et les apprenants (Vygotsky 1962 ; Bruner 1985), mettant en évidence plusieurs fonctions d'étayage ayant pour objectif d'aider les apprenants à résoudre seuls un problème qu'ils ne savaient pas résoudre au préalable. D'autre part, à un autre niveau, des travaux portent sur l'aide aux enseignants pour assister les apprenants (Puustinen & Rouet 2009 ; Richterich 1996). De plus, un certain nombre d'enseignants a déjà mis en place des assistances en contexte éducatif. Ainsi, des enseignants utilisent Hot Potatoes pour créer des exercices et ajoutent des éléments d'assistance à l'utilisateur, comme des indices, un diagnostic, etc. (par exemple dans EOLF). Dans le domaine des MOOC, des enseignants mettent en place de l'assistance, par exemple dans certains MOOC de Fun (FunMooc, 2017). En ce qui concerne SEPIA, l'équipe enseignante du cours d'IHM de licence d'informatique de l'Université Lyon 1 a réalisé cinq tutoriels à l'aide de cet outil pour aider les étudiants à apprendre la programmation avec NetBeans. Ces différents exemples montrent que les enseignants ont des besoins de mise en place d'assistance pour leurs apprenants.

Les assistances définies par les concepteurs d'assistance, les enseignants et les concepteurs des applications sont variées en termes de déroulement. Ainsi, les assistances présentées dans le chapitre 3 se déroulent de manières différentes. Les tutoriels sur NetBeans définis par les

enseignants du cours d'IHM sont proposés aux étudiants sous forme d'étapes d'assistance successives. Dans les étapes dans ces tutoriels, les messages et mises en valeur sont fournies simultanément. Cet exemple montre qu'il existe différents modes de déroulement d'une assistance.

6.2.2 Analyse des processus de transformation en règles aLDEAS

6.2.2.1 Pour le processus d'application automatique des contraintes aux règles

Le processus d'application des contraintes aux règles permet de générer et modifier automatiquement les éléments aLDEAS, ainsi que d'annoncer au concepteur de l'assistance les erreurs qu'il a faites. Les tâches suivantes à réaliser sont faites automatiquement grâce au processus et non plus par le concepteur de l'assistance :

- Pour le mode d'articulation successif entre n règles, n-1 événements « fin d'une règle » sont créés et ajoutés aux n-1 dernières règles. De plus, en cas d'existence d'une condition dans ces n-1 règles, une alternative « sinon » y est ajoutée. Le modèle permet également de détecter les manques d'événements de fin pour les n-1 premières règles.
- Pour le mode d'articulation simultané entre n règles, ces n règles doivent débiter par un événement déclencheur « lancement par une règle ». Ces événements sont ajoutés à ces règles.
- Pour le mode d'articulation interactif entre n règles, ces n règles doivent débiter par un événement déclencheur « lancement par une règle ». Ces événements sont ajoutés automatiquement à ces règles. De plus, une règle R' est créée pour lancer ces règles en mode interactif.
- Le mode d'articulation progressif entre n règles est identique au mode interactif. Cependant, dans ce mode progressif, une consultation du nombre de déclenchements d'une règle créée est ajoutée à la règle R'.
- Le mode d'articulation indépendant entre n règles reflète la définition classique d'une assistance. Il n'y a donc pas de tâche automatisée.

Pour ces différentes tâches automatisées, il est nécessaire qu'un enseignant définisse un bloc de règles dans un mode donné. Cette notion de bloc a désormais dans SEPIA-edu un rôle aussi important de celui des règles. Grâce à cela, les concepteurs peuvent bénéficier d'un nombre de tâches réduit, en particulier pour les modes d'articulations successif et progressif.

La modification et la génération automatique d'éléments aLDEAS, évitent aux concepteurs de créer ces éléments de manière incorrecte. De plus, la détection des manques d'événements de fin pour le mode successif permet d'éviter leur oubli.

6.2.2.2 Pour le processus de transformation du guidage pédagogique en règles aLDEAS

Le processus de transformation du guidage pédagogique en règles aLDEAS permet de générer et de modifier automatiquement les éléments aLDEAS nécessaires. Les tâches

suivantes qu'auraient dû faire les concepteurs sont réalisées automatiquement grâce au modèle. Pour un guidage pédagogique composé de n activités. :

- La première règle « proposition d'activités pertinentes » (cf. Figure 122) est créée. La définition de cette règle est une tâche longue car elle comporte $n + 1$ événements déclencheurs (dont n événements sont des événements « fin d'une règle ») et $n + 1$ actions (dont n actions sont des règles « vérification de l'activité »).
- Les n règles « vérification de l'activité » doivent être également créées. Ces règles contiennent une condition très complexe. Ainsi, cette condition est une combinaison de plusieurs consultations de sources variées (temps, états de sortie, profil). Sa complexité dépend de la complexité du guidage pédagogique.
- Enfin, les n règles « choix de l'activité » doivent être créées. La définition de ces règles n'est pas complexe, toutefois, la définition d'un grand nombre de règles risque toujours de provoquer des confusions ou des erreurs.

L'implémentation du processus dans SEPIA-edu permet d'automatiser ces tâches très complexes. Pour en bénéficier, les concepteurs d'assistance doivent également effectuer des tâches supplémentaires : créer n activités et les placer dans un guidage pédagogique. Cependant, ces tâches sont beaucoup moins lourdes et moins complexes que les tâches automatisées.

6.2.3 Implémentation des modèles et des processus

Nous avons implémenté le modèle d'articulation entre règles et le processus d'application des contraintes aux règles. Afin d'illustrer cette mise en œuvre, nous avons réalisé des assistances variées pour les sites Web Matou Matheux, ActiveMath, EOLF et IXL Learning, ainsi que pour l'application Connectify. Nous montrons la définition et l'exécution de ces assistances par le biais de captures vidéo disponibles en ligne et présentées dans l'annexe D.1 page 215. Dans ces assistances, les blocs respectent le modèle d'articulation correspondant. Grâce au processus d'application des contraintes aux règles, les règles générées et modifiées sont correctement exécutées par le moteur d'assistance.

Nous avons également implémenté le modèle d'activité, le patron de guidage pédagogique, ainsi que le processus de transformation du guidage pédagogique en règles aLDEAS. Afin d'illustrer cette mise en œuvre, nous avons réalisé des guidages pédagogiques variés, notamment sur un MOOC Fun et sur le site MatouMatheux. Nous montrons la définition et l'exécution de ces guidages par le biais de captures vidéo disponibles en ligne et présentées dans l'annexe D.2 page 220. Dans ces guidages, les activités définies respectent le modèle d'activité. De plus, les guidages pédagogiques définis respectent le patron de guidage pédagogique. Les règles générées et modifiées sont correctement exécutées par le moteur d'assistance.

6.2.4 Mise à l'essai de SEPIA-edu en 2015

6.2.4.1 Description

La première mise à l'essai de SEPIA-edu a été conduite en novembre 2015. Nous avons demandé aux 25 étudiants de l'option Logiciels éducatifs du Master d'Informatique de l'Université Lyon 1 d'endosser le rôle de concepteurs d'assistance pour le site Matou Matheux. Ce site Web proposé par l'académie de Rennes propose un grand nombre de ressources pédagogiques en mathématiques mises librement à disposition des enseignants. Le système d'assistance que les étudiants ont contribué à mettre en place visait à adapter des exercices de Matou Matheux pour qu'ils soient en adéquation avec les objectifs donnés d'un enseignant donné pour un contexte donné (adaptation au niveau scolaire, aux élèves, etc.). L'exercice à travailler concernait le thème « les durées et les vitesses ». Dans cet exercice, un élève de niveau 3^{ème} doit convertir une vitesse de km/s en km/h. L'élève doit ensuite donner le résultat sous forme d'une écriture scientifique : sous la forme $a \times 10^b$, en saisissant les valeurs de a et b dans les deux champs correspondants, a étant l'opérande et b étant la puissance de 10.

Cette mise à l'essai avait pour principal objectif l'évaluation de la possibilité, pour des concepteurs d'assistance, de définir un système d'assistance en utilisant un bloc de règles articulées.

6.2.4.2 Déroulement

Chaque étudiant a reçu d'une part un document qui introduisait SEPIA-edu, complété par des vidéos expliquant son fonctionnement, et donnait les étapes à suivre et d'autre part un questionnaire (cf. annexe E.1 page 222).

Plus précisément, chaque participant devait définir une assistance selon trois méthodes de définition différentes :

- uniquement avec des règles en utilisant l'interface textuelle,
- uniquement avec des règles en utilisant l'interface graphique,
- avec des règles et des blocs en utilisant l'interface graphique.

Pour chaque méthode, chaque participant devait réaliser les étapes suivantes :

- compréhension d'un système d'assistance défini avec SEPIA (études des règles et des blocs pour pouvoir décrire le déroulement de l'assistance),
- modification, puis complétion de ce système.

Ensuite, chaque participant devait choisir une méthode pour créer un nouveau système d'assistance.

Enfin, chaque participant devait imaginer un système d'assistance pour un logiciel éducatif. Ils devaient également répondre aux différentes parties du questionnaire au fur à mesure de la séance.

Afin d'évaluer les différents modes d'articulation, les 25 participants étaient regroupés en cinq groupes correspondants aux cinq modes d'articulation entre les règles. Cinq étudiants travaillaient donc pour chaque mode. Pour obtenir un résultat plus objectif, nous avons mélangé l'ordre d'utilisation des méthodes de définition par les participants. Ainsi, nous avons cinq ordres différents des méthodes pour chaque mode d'articulation.

6.2.4.3 Résultats

Après quasiment deux heures de travail, les étudiants ont achevé presque toutes les étapes, à l'exception de l'étape « imaginer un système d'assistance » qui a été fait par un seul étudiant.

La Figure 156 présente le nombre d'étudiants (sur 25 participants) ayant compris le système d'assistance et réalisé la modification de ce système d'assistance selon les trois méthodes de définition.

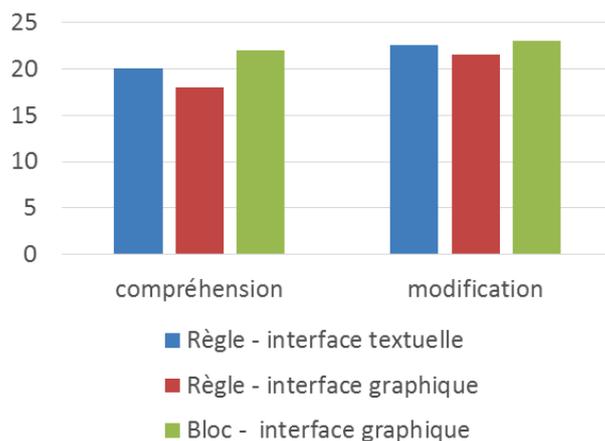


Figure 156 : Résultats relatifs à la réussite pour la compréhension et la modification du système d'assistance selon les trois méthodes de définition

La Figure 157 présente les préférences des participants en matière de méthode de définition d'un nouveau système d'assistance. En particulier, 14 des 25 étudiants ayant choisi la méthode de définition avec des blocs de règles ont réussi créer un nouveau système d'assistance.

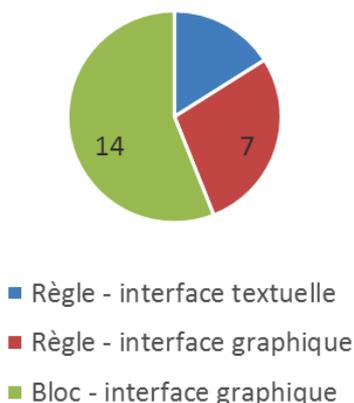
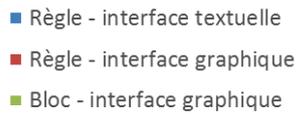
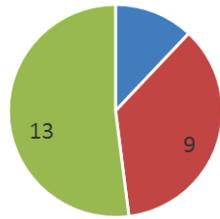


Figure 157 : Résultats relatifs à la préférence les concepteurs pour créer un nouveau système d'assistance

La Figure 158 quant à elle présente les préférences des participants en matière de méthode de définition pour comprendre et définir un système d'assistance. Elle montre la préférence de l'utilisation des blocs par les concepteurs pour ces deux tâches.

Compréhension



Définition

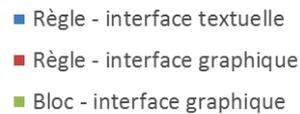
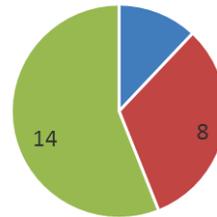
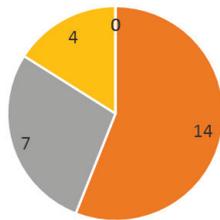


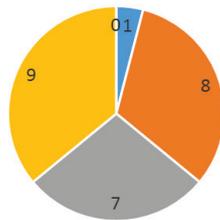
Figure 158 : Résultats relatifs à la préférence d'une méthode par les concepteurs pour comprendre (à gauche) et définir (à droite) un système d'assistance

La Figure 159 (respectivement Figure 160) présente les résultats relatifs à la difficulté de compréhension (respectivement de définition) d'un système d'assistance pour chaque méthode de définition. Ils montrent une difficulté moindre pour le mode bloc.

Règle – interface textuelle



Règle – interface graphique



Bloc – interface graphique

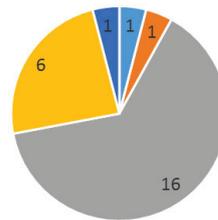
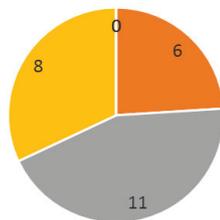
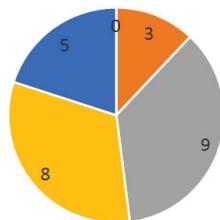


Figure 159 : Résultats relatifs à la difficulté de compréhension d'un système d'assistance

Règle – interface textuelle



Règle – interface graphique



Bloc – interface graphique

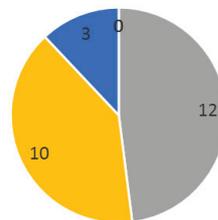


Figure 160 : Résultats relatifs à la difficulté de définition d'un système d'assistance

6.2.4.4 Analyse

Les résultats montrent que les participants peuvent comprendre (22/25), modifier (23/25) et créer (15/25) un système d'assistance avec SEPIA-edu implémentant le modèle d'articulation entre règles.

L'utilisation de SEPIA-edu donne un autre résultat, plus positif encore. En effet, plusieurs participants ont mieux réussi avec les blocs (comprendre (22/25), modifier (23/25), créer (15/25) un système d'assistance) qu'avec les règles avec l'interface textuelle (comprendre (20/25), modifier (22.5/25), créer (8/25) un système d'assistance) ou qu'avec l'interface graphique (comprendre (18/25), modifier (21.5/25), créer (10/25) un système d'assistance). La plupart des participants a choisi l'utilisation des blocs de règles pour créer un nouveau système d'assistance.

Après l'utilisation de SEPIA-edu, la plupart des participants a indiqué sa préférence pour l'utilisation des blocs de règles. De plus, ils ont signalé que l'utilisation des blocs est plus facile. Ainsi, avec la note moyenne relative à la difficulté calculée selon l'échelle : 1 (très difficile), 2 (difficile), 3 (neutre), 4 (facile) et 5 (très facile), l'utilisation des blocs obtient une meilleure note que les autres méthodes : 3.2 pour la compréhension et 3.64 pour la définition. Cependant, l'écart entre les notes des trois méthodes est assez faible.

Plus précisément, presque tous les étudiants qui ont travaillé pour les modes « successif » et « progressif » apprécient l'utilisation des blocs de règles. Une explication à cela est que les blocs de règles articulées dans ces modes facilitent particulièrement la compréhension et la définition d'un système d'assistance complexe. En effet, la consultation « nombre de déclenchements de la règle » et l'événement de fin ne sont faciles ni à comprendre ni à définir. Ainsi, ces éléments sont générés par l'implémentation du modèle d'application automatique des contraintes sur les règles. De plus, ces éléments sont cachés et remplacés par la présentation d'un bloc de règles plus sémantique.

Pour la compréhension des modes d'articulation, presque tous les étudiants ont réussi pour les modes : simultané (25/25), successif (24/25) et progressif (22/25). En revanche, ils sont beaucoup moins nombreux à avoir réussi pour les modes : interactif (15/25) et indépendant (14/25). Ceci s'explique notamment par le fait que certains étudiants ont confondu les noms des modes « interactif » et « indépendant ».

En ce qui concerne la couverture du modèle d'articulation entre règles, les participants n'ont signalé aucun mode d'articulation existant non proposé par SEPIA-edu. Nous avons essayé de renforcer l'évaluation de cette facette en demandant aux étudiants d'imaginer un système d'assistance sur un logiciel éducatif et d'indiquer quels modes d'articulation permettraient de le créer avec SEPIA-edu. Seul un étudiant a effectué cette étape.

6.2.5 Mise à l'essai de SEPIA-edu en 2017

6.2.5.1 Description

Nous avons conduit une dernière mise à l'essai au printemps 2017. Nous avons demandé aux 25 étudiants de l'option Logiciels éducatifs du Master d'Informatique de l'Université Lyon 1 travaillant en binômes d'endosser le rôle d'enseignants du MOOC « Racines ». Ce MOOC Fun

donne des bases d'étymologie à de futurs étudiants de sciences et de santé. Il propose, sur plusieurs semaines, des cours, des exercices et un certain nombre d'outils classiques pour un MOOC (forum, FAQ, index, etc.). Le système d'assistance que contribuaient à mettre en place les étudiants visait à fournir un guidage pédagogique sur ce MOOC pour qu'il soit en adéquation avec les objectifs donnés d'un enseignant donné pour un contexte donné. Ce guidage pédagogique contenait au total cinq activités pour le jour 3 de la semaine 1 du MOOC « Racines ».

Cette mise à l'essai avait pour principal objectif l'évaluation de la possibilité, pour des concepteurs d'assistance, de définir un système d'assistance basé sur la création d'un guidage pédagogique dans SEPIA-edu.

6.2.5.2 Déroulement

Chaque binôme a reçu d'une part un document qui introduisait SEPIA-edu, complété par des vidéos expliquant son fonctionnement, et donnait les étapes à suivre et d'autre part un questionnaire (cf. annexe 0 page 235).

Plus précisément, chaque binôme devait réaliser les étapes suivantes :

- compréhension d'un guidage pédagogique défini avec SEPIA-edu (études des règles, activités et guidage pédagogique pour décrire l'assistance),
- modification du guidage pédagogique.

Ensuite, chaque binôme devait imaginer un guidage pédagogique sur une application informatique et le décrire sur papier en utilisant le patron de guidage pédagogique. Ils devaient également répondre aux différentes parties du questionnaire au fur à mesure de la séance.

6.2.5.3 Résultats

Après quasiment une heure de travail, un seul binôme a réussi à modifier le guidage pédagogique. Les autres binômes sont restés bloqués à la première étape de « compréhension d'un guidage pédagogique défini avec SEPIA-edu ». Un premier problème était lié à l'organisation de test. Ainsi, les étudiants devaient d'une part installer SEPIA-edu sur l'ordinateur de la salle de TP et d'autre part travailler sur le document fourni. Certains binômes n'ont pas su quoi faire en premier. Un deuxième problème était lié à la complexité du document qui comportait des instructions sur le projet AGATE, le système SEPIA-edu, le langage aLDEAS, la notion d'activité, la notion de guidage pédagogique, ainsi que les demandes et le questionnaire pour cette mise à l'essai. Un troisième problème était lié à l'exploitation des possibilités de l'éditeur de SEPIA-edu par des participants. Certains n'ont pas su ouvrir le guidage pédagogique ou manipuler les objets graphiques. Enfin, un dernier problème était lié à l'ergonomie. En effet, l'éditeur de SEPIA-edu comporte un certain nombre d'incohérences entre ses différentes parties (interface graphique pour la définition des règles, des blocs, des guidages pédagogiques). Une amélioration de l'ergonomie de l'interface de SEPIA sera effectuée prochainement. Combinée à une amélioration de la préparation de l'expérimentation, elle devrait rendre possible la mise en place d'une nouvelle instance de cette expérimentation dont nous espérons que les résultats seront satisfaisants.

Chapitre 7. Conclusion et Perspectives

Cette thèse en informatique s'intéresse à l'assistance aux utilisateurs des applications informatiques en contexte éducatif. Dans ce contexte, les utilisateurs que l'on souhaite assister sont des apprenants qui ont besoin d'assistance dans le cadre de leur apprentissage. Les enseignants qui peuvent et souhaitent mettre en place de l'assistance ont donc le rôle de concepteurs d'assistance.

Cette thèse propose une approche unifiée permettant à un enseignant ou un ingénieur pédagogique de mettre en place une assistance technique et pédagogique sur des applications existantes en adoptant une approche épiphyte et générique. L'utilisation de l'outil proposé n'exige pas de compétences en programmation, mais seulement des connaissances pédagogiques. Cela permet aux enseignants de concentrer leurs interventions sur le contenu pédagogique.

Notre approche repose sur des modèles théoriques qui ont été mis en œuvre dans le système SEPIA-edu. Pour conclure ce manuscrit, dans ce chapitre nous présentons une synthèse des travaux réalisés durant cette thèse et nous effectuons un retour sur les scénarios d'usage que nous avons présentés dans l'introduction. Enfin, nous discutons nos contributions, avant de présenter les perspectives que nous envisageons pour des travaux futurs.

Synthèse

Cette thèse a été réalisée dans le cadre du projet AGATE dont les modèles et les outils permettent la mise en place de systèmes d'assistance. Les modèles et outils proposés initialement dans le projet adoptent une démarche générique et épiphyte qui a montré le potentiel de leur utilisation pour des types et des domaines d'application très variés, notamment en contexte éducatif. Cependant, ces outils ne sont pas optimisés pour le contexte éducatif.

Dans cette thèse, nous avons proposé deux principaux apports : d'une part une étude des assistances existantes au sein d'applications utilisées en contexte éducatif, et d'autre part l'exploitation et l'enrichissement des modèles et outils du projet AGATE pour les adapter au contexte éducatif.

Le Chapitre 2 de ce manuscrit est dédié au premier apport et propose une classification des caractéristiques de l'assistance en contexte éducatif. Nous avons complété les caractéristiques des systèmes d'assistance technique identifiées dans la thèse de Blandine Ginon (Ginon, 2014) par des caractéristiques propres à l'assistance pédagogique. En particulier, une assistance pédagogique doit pouvoir fournir des feedbacks pédagogiques qui assistent une activité d'apprentissage et un guidage pédagogique qui organise des activités pour les proposer aux apprenants. De plus, l'assistance peut se dérouler de différentes façons. Nous avons donc également étudié les différents modes de déroulement d'une assistance.

Le Chapitre 3, le Chapitre 4 et le Chapitre 5 de ce manuscrit sont consacrés au deuxième apport et abordent l'exploitation de SEPIA en contexte éducatif. Plus précisément, nous avons confronté le langage aLDEAS permettant de modéliser un système d'assistance en règles aLDEAS et le système SEPIA permettant de définir et d'exécuter ce système d'assistance, aux caractéristiques de l'assistance en contexte éducatif. La confrontation nous amène à aborder deux questions de recherche : « Comment définir le déroulement d'une assistance avec SEPIA ? », et « Comment définir un guidage pédagogique avec SEPIA ? ». Les principales contributions de cette thèse, présentées sous forme synthétique dans la Figure 161, y répondent. Nous revenons à présent sur nos deux questions de recherche pour faire le bilan de nos contributions.

Comment définir le déroulement d'une assistance avec SEPIA ?

Nous avons proposé d'une part un **modèle d'articulation entre règles** (cf. A dans la Figure 161) qui permet d'explicitier les modes de déroulement d'une assistance sous forme de blocs de règles articulées, et d'autre part un **processus d'application des contraintes des modes d'articulation aux règles** (cf. B dans la Figure 161) qui permet d'automatiser certaines tâches préalablement dévolues aux enseignants. Nous avons implémenté ce modèle et ce processus dans le système SEPIA-edu.

Ainsi, notre modèle explicite cinq modes d'articulation entre règles, combinable entre eux : indépendant, successif, simultané, progressif et interactif. Le mode indépendant reflète la définition classique de l'articulation de règles aLDEAS. L'utilisation de ce mode permet de regrouper les règles de même objectif. Les quatre autres modes permettent de définir explicitement l'articulation entre les différents éléments d'un système d'assistance. La mise en place de ces modes impose des contraintes sur les règles pour former une articulation correcte entre elles. Ces contraintes sont représentées dans aLDEAS sous forme d'un patron des règles.

Pour faciliter la définition de règles articulées qui respectent les contraintes des modes d'articulation, nous avons proposé un processus automatique d'application des contraintes des modes d'articulation aux règles. Ce processus permet d'identifier les contraintes des modes d'articulation et les contraintes imposées par d'autres contraintes sur les règles, puis d'appliquer ces contraintes à ces règles. Cette application donne lieu à la génération et/ou la modification d'éléments aLDEAS et/ou la notification des erreurs sur les contraintes non respectées.

L'implémentation du modèle d'articulation a donné lieu dans SEPIA-edu à la création d'une nouvelle étape, destinée aux enseignants, de « définition de blocs de règles articulées » (cf. 1 dans la Figure 161). Les enseignants peuvent donc désormais définir des blocs de règles articulées, ce qui facilite la compréhension d'un système d'assistance, en donnant une vue globale sur les règles.

De plus, les blocs permettent ensuite de générer et/ou de modifier automatiquement les règles grâce à l'implémentation du processus d'application des contraintes (cf. 2 dans la Figure 161). Cette génération et/ou modification permet de maintenir le fonctionnement du moteur préexistant de SEPIA, tout en intégrant les nouvelles possibilités. Les éléments générés sont cachés aux enseignants afin de réduire leur charge de travail.

Bien que les modèles et outils proposés pour répondre à cette question soient issus d'exemples du contexte éducatif, ils restent génériques. Ils peuvent être utilisés pour définir des assistances dans d'autres contextes.

Comment définir un guidage pédagogique avec SEPIA ?

Nous avons proposé un **modèle d'activité** (cf. C dans la Figure 161) qui permet d'identifier les activités existantes des applications-cibles, un **modèle de guidage pédagogique** (cf. D dans la Figure 161) qui permet d'explicitier les types de guidages pédagogiques, un **patron de guidage pédagogique** (cf. E dans la Figure 161) qui permet de définir un guidage pédagogique complexe qui peut combiner différents types de guidages pédagogiques et un **processus de transformation du guidage pédagogique en règles aLDEAS** (cf. F dans la Figure 161) qui automatise certaines tâches auparavant réalisées par les enseignants. Nous avons implémenté ces modèles et processus dans SEPIA-edu, à l'exclusion du modèle de guidage pédagogique

qu'il n'a pas été nécessaire d'implémenter car le patron de guidage pédagogique permet également de définir les différents types de guidages.

Le modèle d'activité permet d'identifier les activités des applications-cibles dans SEPIA-edu. Ce modèle ne fournit pas toutes les informations sur les activités, mais seulement les informations indispensables à leur exécution dans le guidage pédagogique de SEPIA-edu. Dans ce modèle, en plus du nom et de la description de l'activité, les enseignants doivent identifier d'une part les états de sortie de l'activité selon le formalisme PMDL et d'autre part les règles aLDEAS associées à cette activité qui détectent la fin d'une activité et identifient ses états de sortie et qui assistent l'apprenant dans cette activité.

Le modèle de guidage pédagogique permet d'explicitier cinq types de guidages pédagogiques : libre, séquentiel, temporel, personnel et contextuel. Il donne également la traduction en aLDEAS de chaque type de guidage pédagogique. Cependant, en pratique, un guidage pédagogique se fait généralement par la combinaison de plusieurs types de guidages pédagogiques. Nous avons donc proposé un patron de guidage pédagogique qui permet de définir des guidages pédagogiques variés. Associée à ce patron, nous avons proposé une traduction des guidages pédagogiques en règles aLDEAS.

L'implémentation du modèle d'activité a donné lieu à la mise en place d'une nouvelle étape destinée aux enseignants dans SEPIA-edu : l'« identification des activités » (cf. 3 dans la Figure 161). Les enseignants peuvent définir des activités avec des états de sortie et les règles associées (regroupées en blocs). Les enseignants peuvent ainsi savoir explicitement quelles règles font partie de quelles activités et quels états sont associés à quelle activité. Au moment de la définition d'une activité, les enseignants n'ont pas besoin de savoir comment cette activité est liée aux autres activités, comment elle est lancée lors de la fin d'une autre activité et comment une autre activité est lancée après cette activité. Toujours pour faciliter le travail des enseignants, nous avons également proposé et implémenté un mécanisme d'activation et de désactivation des règles associées aux activités. Ce mécanisme permet d'activer les règles qui sont associées à l'activité en cours ou celles associées à aucune activité. À présent, seules les règles activées pourront être lancées lors des événements déclencheurs, alors que précédemment toutes les règles d'un système d'assistance pourraient être lancées.

Le patron de guidage pédagogique permet de définir un guidage pédagogique complet qui combine les cinq types de guidages pédagogiques. Son implémentation dans SEPIA-edu a donné lieu à la mise en place d'une nouvelle étape, destinée aux enseignants, de « définition du guidage pédagogique » (cf. 4 dans la Figure 161). Les enseignants peuvent désormais définir un guidage pédagogique et comprendre comment les activités s'organisent en examinant ce guidage qui donne une vue sur cette organisation. Ils peuvent ensuite facilement modifier ce guidage pédagogique.

De plus, le guidage pédagogique permet de générer et/ou de modifier les règles grâce à l'implémentation du processus de transformation d'un guidage pédagogique en règles aLDEAS (cf. 5 dans la Figure 161). Cette génération et/ou modification permet de maintenir le fonctionnement préexistant du moteur de SEPIA-edu. Les éléments aLDEAS générés et modifiés sont cachés aux enseignants. Ainsi, à la fin de la définition du système d'assistance, un fichier qui contient tous les éléments (y compris les éléments générés et/ou modifiés) qui définissent le système d'assistance est généré et est ensuite exécuté par le moteur de SEPIA-edu.

Retours sur les scénarios d'usage

Nous revenons maintenant sur les scénarios d'usage présentés en introduction, afin de montrer dans quelle mesure nos propositions répondent aux besoins des acteurs dans ces scénarios.

Scénario 1 : proposition d'un retour pédagogique pour un exercice de Matou Matheux

Stéphane souhaite créer un système d'assistance qui fournirait des feedbacks de façon progressive directement à l'interface de Matou Matheux. Cela pourrait prendre la forme d'un bouton d'aide sur lequel l'apprenant cliquerait pour demander de l'assistance lorsqu'il rencontre une difficulté. Lors de la première demande d'aide d'un élève, une explication serait proposée sous forme d'un message. Lors de la deuxième demande d'aide sur le même élément, un exemple serait proposé sous forme de messages et de flèches pointant sur les composants concernés de l'interface. Ces messages et flèches seraient affichés les uns après les autres.

Avec SEPIA-edu, Stéphane peut maintenant créer un ensemble de règles qui définissent ce système d'assistance. En utilisant un bloc de règles en mode progressif, il peut définir facilement et explicitement le déroulement progressif des feedbacks pédagogiques. Ainsi, il n'a pas besoin de savoir comment créer une consultation du « nombre de déclenchements d'une règle » dans l'historique de l'assistance, car elle est générée automatiquement. De plus, à l'avenir, il pourra comprendre rapidement le déroulement de ce système en examinant le bloc qu'il a défini et éventuellement changer le mode de ce bloc. Nous avons créé et exécuté le système d'assistance correspondant à ce scénario, il est présenté dans l'annexe D.1 page 215.

Scénario 2 : compléter l'activité d'apprentissage de GeoGebra

Stéphane souhaite créer dans le site Web de GeoGebra un système d'assistance comportant des instructions sous forme de messages successifs. Ces instructions expliciteraient pour les apprenants chaque étape de construction des objets géométriques : construction d'une première droite sans contrainte, puis construction de deux autres droites perpendiculaires à la première, pour ensuite pouvoir démontrer que les deux dernières droites sont parallèles.

Avec SEPIA-edu, Stéphane peut maintenant créer un ensemble de règles qui définissent ce système d'assistance. En utilisant un bloc de règles en mode successif, il peut définir facilement et explicitement le déroulement successif des messages. Ainsi, il n'a pas besoin de savoir comment créer les événements de « fin d'une règle », car ils sont générés automatiquement. Il est invité à ajouter les événements de fin manquants sur certaines règles pour assurer le bon déroulement de l'assistance. Le système d'assistance exécuté complète l'activité proposée sur GeoGebra en fournissant à l'apprenant les instructions explicitant l'activité. Nous avons créé et exécuté le système d'assistance correspondant à ce scénario, il est présenté dans l'annexe D.1 page 215.

Scénario 3 : proposition d'une activité sur NetBeans

Hélène souhaite créer un système d'assistance dans lequel un tutoriel est affiché directement sur l'interface de NetBeans. Ce tutoriel se composerait de messages et de flèches pour guider les étudiants dans la création d'un projet NetBeans, la manipulation de ses propriétés, la création d'une fenêtre simple et la création de fonctions gérant les événements

à l'interface. Ce tutoriel se déclencherait en fonction des actions des apprenants à l'interface de NetBeans.

Avec SEPIA-edu, Hélène peut maintenant créer un ensemble de règles qui définissent ce système d'assistance. Comme Stéphane dans le scénario précédent, Hélène utilise un bloc de règles en mode successif pour faciliter son travail. Le système d'assistance exécuté propose une activité sous forme d'un tutoriel qui permet aux étudiants de débiter la programmation événementielle avec NetBeans. Ce scénario a été mis en œuvre lors de l'expérimentation de 2014 (Ginon, Thai, *et al.*, 2014), il pourrait désormais être mis en œuvre de façon plus facile pour les concepteurs d'assistance et plus pertinente pour les étudiants avec SEPIA-edu. Nous avons créé et exécuté le système d'assistance correspondant à ce scénario, il est présenté dans l'annexe D.1 page 215.

Scénario 4 : guidage pédagogique au sein d'une plateforme d'exercices d'auto-évaluation

Marie souhaite créer un système d'assistance qui fournit un guidage pédagogique pour afficher une liste d'activités pertinentes à réaliser par ses étudiants. Cette liste est définie en fonction de la progression de chaque étudiant au sein de la plateforme. Plus précisément, deux premières activités simples pourraient être présentées aux apprenants, puis deux autres activités seraient proposées seulement si l'étudiant termine les deux premières sans les réussir du premier coup. Un tel guidage pédagogique assurerait une bonne progression des étudiants. Ce système d'assistance comportant un guidage pédagogique, créé une année, pourrait être réutilisé et amélioré pour les années suivantes.

Avec SEPIA-edu, Marie peut maintenant créer un ensemble de règles qui définissent ce système d'assistance. Elle peut définir facilement et explicitement ce guidage pédagogique. Ainsi, elle peut définir dans SEPIA-edu 12 activités, chacune avec un état de sortie réussite/échec, ainsi qu'un guidage pédagogique dans lequel elle peut organiser ces activités. Elle n'a pas besoin de savoir comment définir les règles formant un guidage pédagogique, qui sont nombreuses pour 12 activités. De plus, à l'avenir, elle pourra comprendre rapidement le système d'assistance en examinant le guidage pédagogique et les activités qu'il contient. Le système d'assistance exécuté propose une liste d'activités pertinentes aux étudiants pour qu'ils choisissent celles qu'ils souhaitent réaliser. Nous avons créé et exécuté le système d'assistance correspondant à ce scénario, il est présenté dans l'annexe D.2 page 220.

Discussion

Les modèles et les processus proposés et implémentés dans SEPIA-edu durant cette thèse établissent l'articulation entre la généralité d'un assistant intelligent et les spécificités de son usage en contexte éducatif. Ainsi, ils permettent d'ajouter au système SEPIA une surcouche par rapport au fonctionnement par règles (cf. Figure 162) : la définition de blocs de règles articulées selon un mode donné, l'identification des activités de l'application-cible, la définition des états de sortie de ces activités et la définition d'un guidage pédagogique. Cette surcouche ne perturbe pas le fonctionnement précédent de SEPIA, mais permet d'adapter SEPIA au contexte éducatif.

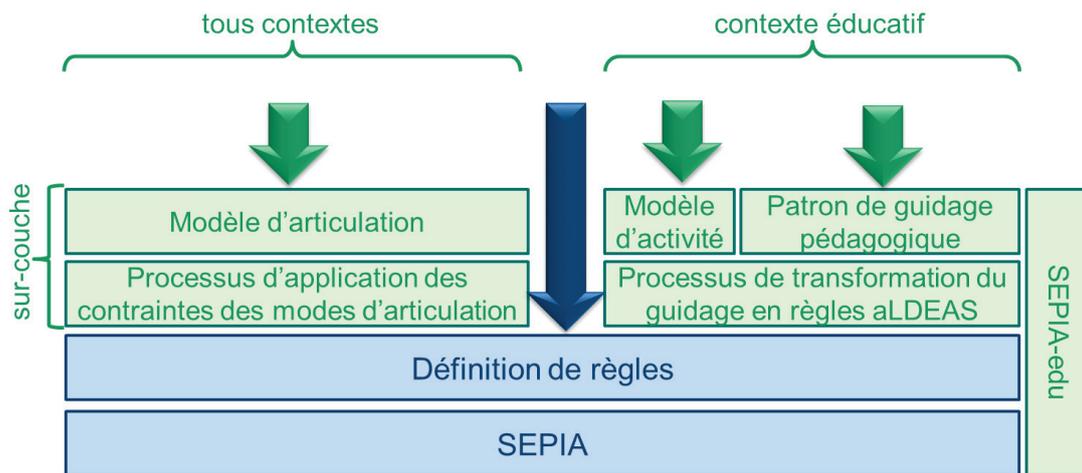


Figure 162 : Système SEPIA-edu constitué par le système SEPIA et la surcouche métier

Ainsi, cette surcouche métier ajoutée à SEPIA-edu est plus adaptée aux enseignants que le fonctionnement uniquement par règles. En effet, elle intègre des éléments qui comportent une dimension sémantique afin de prendre en compte l'aspect pédagogique : mode d'articulation, activité, état de sortie, guidage pédagogique. Dans SEPIA-edu, les enseignants peuvent se concentrer plus sur ces éléments sémantiques et moins sur les règles, qui sont plus abstraites. Cette surcouche explicite et facilite donc la définition d'un système d'assistance par les enseignants.

De plus, cette surcouche au système SEPIA-edu ne change pas le fonctionnement de base de SEPIA : les systèmes d'assistance se définissent toujours par un ensemble de règles aLDEAS. Le bloc, l'activité, le guidage pédagogique contenus dans un système d'assistance sont convertis en règles aLDEAS grâce aux processus automatiques de transformation en règles aLDEAS. Cela permet donc de maintenir la compatibilité du système d'assistance défini avec le moteur de SEPIA-edu.

Cette surcouche est issue du contexte éducatif, mais certains éléments peuvent être exploités dans différents contextes. Ainsi, l'articulation entre règles peut être exploitée pour tous les contextes, alors que le guidage pédagogique est exploité principalement en contexte éducatif (cf. Figure 162). Le guidage pédagogique peut toutefois être également exploité dans d'autres contextes par les utilisateurs de SEPIA-edu qui souhaitent planifier des activités dans l'application-cible. Par exemple, dans le contexte industriel, le chef de projet peut planifier les activités pour des membres du projet à l'aide d'un guidage pédagogique.

Malgré ces points forts, cette surcouche n'est pas une solution parfaite, notamment en raison de son interface améliorable. L'amélioration de l'ergonomie est prévue à court terme.

De plus, la définition du guidage pédagogique serait plus flexible si SEPIA-edu fournissait des représentations plus variées d'un guidage pédagogique (par exemple, une représentation temporelle permettrait de placer les activités sur un axe du temps).

D'autres limites de notre travail sont issues du cadre d'application de cette thèse, qui découle de celui du système SEPIA. Il s'agit de limites techniques concernant des composants des applications-cibles : composants dynamiques de l'interface et composants inaccessibles de l'interface. Ces limites n'empêchent toutefois pas totalement la création de systèmes d'assistance sur de telles applications-cibles. Au niveau d'une activité, ces systèmes peuvent ne

pas permettre une assistance complètement contextualisée. Cependant, pour contourner ces difficultés, l'assistance peut être fournie à la demande de l'apprenant. L'apprenant peut exprimer ses problèmes ou ses difficultés *via* des consultations de l'apprenant par le système d'assistance, afin que le système puisse fournir l'assistance adaptée. Par exemple, un système d'assistance fournit les instructions explicitant une activité de création d'objets géométriques avec GeoGebra. La zone de construction des objets géométriques de GeoGebra n'est pas accessible par SEPIA, car non ouverte aux techniques d'accessibilité pour le handicap. Une instruction spécifique ne peut donc pas être fournie lors d'une action sur cette zone par l'apprenant. Mais l'instruction peut être fournie lors d'une confirmation par l'apprenant que cette action a été réalisée. Au niveau du guidage pédagogique, les limites ne concernent pas l'accessibilité des composants, mais l'accessibilité des états de sortie des activités, le profil de l'apprenant, etc.

En plus de ces limites techniques qui restent à résoudre, nous avons identifié des problématiques de recherche que nous aimerions traiter à l'avenir. Elles sont présentées dans la section suivante.

Perspectives

Prise en compte du caractère bidirectionnel de l'assistance en mode successif

Actuellement, le mode d'articulation successif permet de définir le déroulement d'une assistance successive dans une seule direction. C'est-à-dire que les éléments d'assistance sont fournis les uns après les autres et que les utilisateurs ne peuvent pas revenir aux éléments précédents. Pourtant, revenir en arrière sur l'assistance pour mieux en comprendre un élément peut être utile pour les utilisateurs de l'assistance, particulièrement en contexte éducatif.

Comme perspective à court terme, nous souhaitons proposer un mode d'articulation **successif bidirectionnel**. Pour cela, nous devons faire évoluer le modèle d'articulation entre règles et le processus d'application automatique des contraintes des modes d'articulation. Nous devons également faire évoluer SEPIA-edu qui implémente ce modèle et ce processus.

Représentation globale des blocs de règles

SEPIA-edu propose une visualisation graphique d'un bloc. Cependant, les modes d'articulation entre règles peuvent être combinés et cette combinaison n'est pas complètement explicite. Les concepteurs ne peuvent pas avoir une vue globale d'un ensemble de blocs combinés *via* la représentation d'un bloc unique. De plus, la définition de la combinaison des blocs peut être difficile car les enseignants doivent définir séparément ces blocs et faire basculer l'affichage entre les blocs.

Comme perspective à court terme, nous visons à proposer une représentation globale des blocs des règles. Cette représentation pourra nécessiter la création d'un patron pour définir la combinaison des blocs de règles et son implémentation dans SEPIA-edu au travers d'une interface graphique.

Prise en compte du niveau de priorité des activités dans le guidage pédagogique

Dans un guidage pédagogique, certaines activités peuvent être prioritaires par rapport à d'autres en fonction des besoins des enseignants. Par exemple, une activité non encore réalisée

devrait pouvoir être recommandée prioritairement aux autres activités. Cependant, avec SEPIA-edu, un guidage pédagogique organise les activités sans notion de priorité. Ainsi, le guidage pédagogique permet aux concepteurs d'assistance de spécifier quelles activités il juge pertinentes pour les apprenants. Dans ce guidage pédagogique, les activités proposées sont toutes au même niveau.

Actuellement, une façon pour les concepteurs de prioriser les activités dans SEPIA-edu est de définir un guidage séquentiel qui permet de "déverrouiller" les activités les plus prioritaires en premier. Cependant, cela contraint fortement le choix des activités par l'apprenant. Ainsi, un enseignant ne peut pas "conseiller" les activités prioritaires par rapport à d'autres, il doit "obliger" à faire les activités prioritaires avant les autres activités. Nous souhaiterions apporter plus de souplesse par la notion d'activités prioritaires.

Comme perspective à moyen terme, nous aimerions donc ajouter à nos modèles et à notre système la prise en compte du niveau de priorité des activités dans le guidage pédagogique. Pour cela, nous devons étudier les sources qui peuvent contraindre la priorité des activités, avant de pouvoir soit faire évoluer nos modèles, soit proposer de nouveaux modèles prenant en charge cette notion.

Assistance à l'enseignant dans la définition de l'assistance à l'apprenant

Actuellement, SEPIA-edu permet aux enseignants de définir les règles, les blocs, les activités et le guidage pédagogique nécessaires à la spécification d'une assistance pédagogique destinée à leurs apprenants. Cependant, c'est une tâche difficile et itérative (l'enseignant peut souhaiter faire évoluer un système d'assistance après des retours suite à l'utilisation des apprenants). Il se peut que certains enseignants ne sachent pas comment créer un système d'assistance ou que les systèmes d'assistance qu'ils créent ne soient pas suffisamment efficaces.

Comme perspective à long terme, nous souhaitons étudier finement les sources de ces difficultés. Pour cela, nous envisageons deux pistes :

- **Prise en compte des retours des usages par les apprenants des systèmes d'assistance conçus par les enseignants.** Les retours sur l'usage des différentes composantes de l'assistance (feedbacks pédagogiques, activités, guidage pédagogique ou déroulement de l'assistance) pourraient permettre aux enseignants de corriger leur système d'assistance. Par exemple, il serait intéressant d'établir des indicateurs permettant de répondre à la question « les explications permettent-elles aux apprenants de comprendre leurs erreurs ? ».
- **Proposition d'une assistance SEPIA guidant le concepteur de l'assistance dans la définition du guidage pédagogique dans SEPIA-edu.** Cette piste est destinée principalement aux enseignants qui n'ont pas assez d'expérience dans la création d'un guidage pédagogique. Cette assistance à SEPIA-edu devra prendre en compte les besoins de l'enseignant pour, d'une part générer le guidage pédagogique, les activités avec ou sans état de sortie, les règles correspondantes, et d'autre part fournir les instructions complémentaires pour achever la définition du guidage, des activités, des états de sortie et des règles. Une telle assistance ne peut pas être réalisée avec la version actuelle de SEPIA-edu, cela nécessiterait en effet, la prise en compte des spécificités des besoins des enseignants et de la situation de classe, ainsi que la proposition de patrons d'utilisation de SEPIA-edu.

Prise en compte de différents niveaux de granularité dans le guidage pédagogique

Dans le cadre de cette thèse, nous nous sommes intéressés au niveau activités, mais pas au niveau supérieur de structuration des activités (formation, module, cours, leçon, etc.) (Pernin 2004). La prise en compte de tels niveaux dans un guidage pédagogique permettrait de donner une solution complète à plus grande échelle (celle d'une classe, d'une école, d'un département, etc.). Le guidage pédagogique guiderait donc non seulement vers un ensemble d'activités, mais également vers un ensemble de modules ou de cours.

De plus, dans SEPIA-edu, nous avons adopté le point de vue d'un concepteur d'assistance, l'enseignant en contexte éducatif, concevant seul l'aide pour ses apprenants. Le système n'a pas été imaginé pour une définition collaborative de l'assistance. Ainsi, les enseignants et les tuteurs d'un même cours pourraient chacun proposer leurs propres activités dans un guidage pédagogique séparé. Pour proposer un guidage pédagogique unique qui prenne en compte l'ensemble de leurs activités, ces enseignants doivent pour le moment travailler ensemble sur un même ordinateur dans un même fichier d'assistance. Il n'y a pas de notion d'utilisateur associée aux activités, ce qui peut augmenter la charge de travail des enseignants. De plus, l'enseignant responsable du cours pourrait proposer une activité à un niveau de granularité fort, que le tuteur pourrait détailler en activités de granularité plus fine. De telles sous-activités ne sont pas encore prises en compte dans SEPIA-edu.

Comme perspective à long terme, nous aimerions permettre en place une gestion de ces différents niveaux de granularité dans la conception d'un système d'assistance, ainsi qu'une gestion de différents concepteurs qui contribuent au système d'assistance. C'est un travail complexe qui nécessitera des changements importants à la fois dans les modèles du projet et dans le système SEPIA-edu.

Intégration des connaissances du domaine pour permettre un diagnostic des réponses de l'apprenant avec SEPIA-edu :

Le langage aLDEAS et sa mise en œuvre dans SEPIA-edu permettent d'ores et déjà de décrire un diagnostic des réponses des apprenants. Ainsi, la consultation de l'état de l'application-cible permet de recueillir les réponses de l'apprenant. Le diagnostic se fait par la comparaison de ces réponses aux réponses correctes et aux réponses incorrectes attendues. Cependant, l'enseignant doit préciser toutes les réponses possibles pour fournir un diagnostic pertinent, ce qui n'est pas envisageable dans certains cas.

Pour illustrer les limites de la mise en place d'un diagnostic des réponses dans SEPIA-edu, prenons l'exemple d'un enseignant qui souhaite mettre en place un diagnostic dans une application-cible qui vise la résolution d'équations. L'apprenant doit répondre à une question en trouvant la valeur de X dans l'équation « $(X + 1) + (X - 2) = 0$ ». Pour cette question, l'enseignant identifie trois types de réponses possibles correspondant à trois messages qui seront affichés à l'apprenant selon la situation : « Réponse correcte » ($X = 1$), « Équation équivalente, mais ce n'est pas la réponse finale » (par exemple, $X - 1 = 0$ ou $-X = -1$), « Réponse incorrecte » (par exemple, $X = 2$ ou $X - 2 = 0$). Pour définir ce diagnostic avec SEPIA-edu, l'enseignant crée une règle (cf. Figure 163) qui consulte la réponse de l'apprenant en récupérant la valeur de la zone de saisie correspondante sur l'interface. Cette réponse est comparée aux trois réponses possibles : $X = 1$, $X - 1 = 0$ et $X = 2$ pour afficher le message adapté. Cependant, cette règle ne peut pas couvrir toutes les réponses possibles de l'apprenant. Il est souhaitable que le système d'assistance puisse considérer, dans certaines situations

(dépendant de l'objectif pédagogique de l'enseignant), que la réponse « $X - 1 = 0$ » est équivalente à la réponse « $-X = -1$ ».

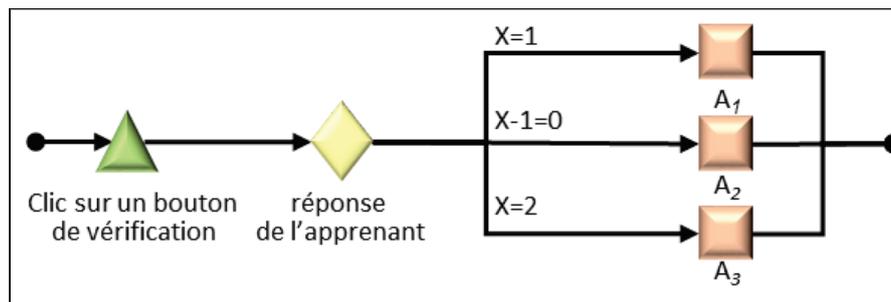


Figure 163 : Exemple de règle permettant un diagnostic dans SEPIA

L'utilisation de SEPIA-edu pour mettre en place du diagnostic devient donc très rapidement complexe, voire impossible si l'enseignant souhaite un diagnostic complet, puisqu'il ne peut pas préciser toutes les potentielles réponses des apprenants dans une règle.

Mettre en place un tel diagnostic complexe dans SEPIA-edu, tout en maintenant la généralité de l'approche, nécessiterait l'intégration de connaissances du domaine. Ainsi, dans le cadre d'un autre travail de recherche du projet AGATE, Margot Fargier (Fargier, 2016) a posé les bases d'un processus de diagnostic des réponses algébriques des apprenants. De premiers éléments de diagnostic ont pu être mis en œuvre par l'intégration de connaissances du domaine, des connaissances algébriques, dans SEPIA-edu. Cette intégration a été faite grâce à l'exploitation du résolveur Giac (Parisse and Graeve, 2016).

Pour que SEPIA-edu soit pleinement exploitable dans le domaine éducatif, il est nécessaire d'étendre ces premiers résultats sur le diagnostic des réponses de l'apprenant et donc d'intégrer à SEPIA-edu des connaissances d'autres domaines. Cette intégration pourra se faire de plusieurs manières : en intégrant des connaissances du domaine existantes et validées dans des logiciels et/ou validées par la communauté ; mais elle pourra également se faire *via* une acquisition interactive de connaissances avec les utilisateurs de SEPIA-edu.

Références

(AGATE, 2016)

AGATE 2016 *Le projet AGATE*. Available at: <http://liris.cnrs.fr/agate/>.

(Antoniadis *et al.*, 2005)

Antoniadis, G., Kraif, O., Lebarbé, T., Ponton, C. and Echinard, S. 2005 'Modélisation de l'intégration de ressources TAL pour l'apprentissage des langues: la plateforme MIRTO', *ALSIC. Apprentissage des Langues et Systèmes d'Information et de Communication*. Adalsic, 8(2).

(Aplusix, 2016)

Aplusix 2016 *Aplusix*. Available at: <http://www.aplusix.com/fr/>.

(Balla *et al.*, 2004)

Balla, A., Hidouci, K.-W., Ihadadene, N. and Hanouh, A. 2004 'Un modèle de système pédagogique adaptatif', in *Colloque international TICE Méditerranée*.

(Bouhineau *et al.*, 2002)

Bouhineau, D., Nicaud, J.-F., Pavard, X. and Sander, E. 2002 'A Microworld for Helping Students to Learn Algebra', *proceedings of ICTMT*, 5, p. 9--pages.

(Bruner, 1985)

Bruner, J. 1985 'Child's talk: Learning to use language', *Child Language Teaching and Therapy*. SAGE Publications Sage UK: London, England, 1(1), pp. 111–114.

(Brusilovsky, 2001)

Brusilovsky, P. 2001 'Adaptive Hypermedia', *User Modeling and User Adapted Interaction*, 11, pp. 87–110.

(Camtasia, 2016)

Camtasia 2016. Available at: <http://www.techsmith.fr/camtasia.html>.

(Carlier and Renault, 2010)

Carlier, F. and Renault, V. 2010 'Educational webportals augmented by mobile devices with iFrimousse architecture', in *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pp. 236–240.

(Connectify, 2016)

Connectify 2016. Available at: <http://www.connectify.me/>.

(Cordier *et al.*, 2010)

Cordier, A., Lefevre, M., Jean-Daubias, S. and Guin, N. 2010 'Concevoir des assistants intelligents pour des applications fortement orientées connaissances : problématiques, enjeux et étude de cas', in *Acte des 21èmes Journées Francophones d'Ingénierie des Connaissances*, pp. 119–131. Available at: <http://hal.archives-ouvertes.fr/hal-00487659%5Cnhttp://liris.cnrs.fr/publis/?id=4642>.

- (Cortes Buitrago, 1999)
Cortes Buitrago, G. 1999 Simulations et contrôle pédagogique: architectures logicielles réutilisables. Grenoble 1.
- (David, 1995)
David, R. 1995 'Grafcet: A powerful tool for specification of logic controllers', *Control Systems Technology, IEEE Transactions on*. IEEE, 3(3), pp. 253–268.
- (Djoudi and Behaz, 2016)
Djoudi, M. and Behaz, A. 2016 'Génération dynamique de documents hypermédias adaptatifs dans un environnement numérique de travail', *REVUE AFRICAINE DE LA RECHERCHE EN INFORMATIQUE ET MATHÉMATIQUES APPLIQUÉES*. Episciences. org, 3.
- (EchosEcole, 2017)
Échos d'école 2017. Available at: <https://echosdecole.com>.
- (EducaSource, 2017)
EducaSource 2017. Available at: <http://www.educasources.education.fr/>.
- (EDUCLEVER, 2016)
EDUCLEVER 2016. Available at: <http://www.educlever.com/fr/>.
- (EOLF, 2016)
EOLF 2016. Available at: <http://eolf.univ-fcomte.fr/>.
- (Ergun, 2006)
Ergun, M. 2006 Etude didactique d'une activité de conception de protocole expérimental en chimie, supportée par un environnement informatique dédié. Etude de cas d'Educ@ ffix.net. Université Joseph-Fourier-Grenoble I.
- (Etienne, 1987)
Etienne, W. 1987 'Artificial intelligence and tutoring systems', *Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann, Los Altos, San Francisco, CA USA.
- (Eyssautier-Bavay, 2008)
Eyssautier-Bavay, C. 2008 Modèles, langage et outils pour la réutilisation de profils d'apprenants. Université Joseph-Fourier-Grenoble I.
- (Falstad, 2017)
Falstad 2017. Available at: <http://www.falstad.com/>.
- (Fargier, 2016)
Fargier, M. 2016 Agate-Idok : Intégration de Connaissances du Domaine à Sepia pour Améliorer les Systèmes d'Assistance Épiphytes Produits.
- (FunMooc, 2017)
FunMooc 2017 *Mooc de fun*. Available at: <https://www.fun-mooc.fr/>.
- (Galisson and Nouveau, 2003)
Galisson, A. and Nouveau, J. S. 2003 'OASIF: un outil collaboratif d'aide à la scénarisation de modules de formation ouverte et à distance', *Télécom Paris, Department of Educational Innovation, France*.
- (GeoGebra, 2016)
GeoGebra 2016. Available at: <https://www.geogebra.org/m/S5zFhkPm>.

- (Ginon, 2014)
Ginon, B. 2014 *Modèles et outils génériques pour mettre en place des systèmes d'assistance épiphytes*. INSA de Lyon, Université de Lyon. Available at: <http://liris.cnrs.fr/publis/?id=6885>.
- (Ginon, Champin and Jean-Daubias, 2013)
Ginon, B., Champin, P.-A. and Jean-Daubias, S. 2013 'Collecting fine-grained use traces in any application without modifying it', in *workshop EXPPORT from the conference ICCBR*, pp. 1–10. Available at: <http://liris.cnrs.fr/publis/?id=6141>.
- (Ginon, Jean-Daubias and Champin, 2013)
Ginon, B., Jean-Daubias, S. and Champin, P.-A. 2013 'Mise en place d'un système d'assistance personnalisée dans une application existante', in *Ingénierie des connaissances*, pp. 137–148. Available at: <http://liris.cnrs.fr/publis/?id=6238>.
- (Ginon, Jean-Daubias, et al., 2014a)
Ginon, B., Jean-Daubias, S., Champin, P.-A. and Lefevre, M. 2014a 'aLDEAS: a Language to Define Epiphytic Assistance Systems ', in *EKAW*, pp. 153–164. doi: 10.1007/978-3-319-13704-9_12.
- (Ginon, Jean-Daubias, et al., 2014b)
Ginon, B., Jean-Daubias, S., Champin, P.-A. and Lefevre, M. 2014b 'Setup of epiphytic assistance systems with SEPIA', pp. 1–4. Available at: <http://liris.cnrs.fr/publis/?id=6990>.
- (Ginon, Thai, et al., 2014)
Ginon, B., Thai, L.-V., Jean-Daubias, S., Lefèvre, M. and Pierre-Antoine, C. 2014 'Adding epiphytic assistance systems in learning applications using the SEPIA system', in *EC-TEL*.
- (GoMaths, 2017)
GoMaths 2017. Available at: <http://www.gomaths.ch/fractions.php>.
- (Google, 2016)
Google 2016. Available at: <https://support.google.com/chrome/>.
- (Gounon, Dubourg and Leroux, 2004)
Gounon, P., Dubourg, X. and Leroux, P. 2004 'Un modèle d'organisation du tutorat pour la conception de dispositifs informatiques d'accompagnement des apprenants', in *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie*, pp. 369–376.
- (Gounon, Leroux and Dubourg, 2004)
Gounon, P., Leroux, P. and Dubourg, X. 2004 'Proposition d'un modèle de tutorat pour la conception de dispositifs d'accompagnement en formation en ligne', *Revue internationale des technologies en pédagogie universitaire*, 1(3), pp. 14–33.
- (Guéraud, 2006)
Guéraud, V. 2006 'Une approche auteur pour les scénarios d'activités', *Scénariser l'enseignement et l'apprentissage: une nouvelle compétence pour le praticien*, pp. 33–38.
- (Guéraud et al., 2004)
Guéraud, V., Adam, J.-M., Pernin, J.-P., Calvary, G. and David, J.-P. 2004 'L'exploitation d'Objets Pédagogiques Interactifs à distance: le projet FORMID', *Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, 11, p. 46--pages.

- (Guéraud *et al.*, 2009)
Guéraud, V., Lejeune, A., Adam, J.-M., Dubois, M. and Mandran, N. 2009 'Supervising distant simulation-based practical work: environment and experimentation', in *European Conference on Technology Enhanced Learning*, pp. 602–608.
- (Gymglish, 2016)
Gymglish 2016 *Gymglish*. Available at: <https://www.gymglish.com/fr/>.
- (Harper, Khan and Stevens, 2005)
Harper, S., Khan, G. and Stevens, R. 2005 'Design checks for Java accessibility', *Accessible Design in the Digital World, UK*. Citeseer.
- (Haverty, 2005)
Haverty, R. 2005 'New accessibility model for Microsoft Windows and cross platform development', *ACM SIGACCESS Accessibility and Computing*. ACM, (82), pp. 11–17.
- (Hotte, Godinet and Pernin, 2007)
Hotte, R., Godinet, H. and Pernin, J.-P. 2007 'Scénariser l'apprentissage, une activité de modélisation', *Revue internationale des technologies en pédagogie universitaire*, 4(2), pp. 7–13.
- (Imagememo, 2017)
Imagememo 2017. Available at: <http://www.imagememo.fr/>.
- (IMS-LD, 2016)
IMS-LD 2016 IMS Learning Design. Information Model, Best Practice and Implementation Guide, Binding document, Schemas. Available at: <http://www.imsglobal.org/learningdesign/index.html>.
- (IXL, 2016)
IXL 2016 *IXL Learning*. Available at: <https://eu.ixl.com/>.
- (Jean-Daubias, 2011)
Jean-Daubias, S. 2011 'TRI, un logiciel de sensibilisation au tri sélectif et au recyclage s'appuyant sur des sessions personnalisables', in *EIAH 2011-Environnements Informatique pour l'Apprentissage Humain-Atelier' Personnalisation de l'apprentissage: quelles approches pour quels besoins?*, pp. 1–8.
- (Jean-Daubias, Ginon and Lefevre, 2011)
Jean-Daubias, S., Ginon, B. and Lefevre, M. 2011 'Modèles et outils pour prendre en compte l'évolutivité dans les profils d'apprenants', *Revue Sticef. org*, 16, p. 26.
- (Koper, 2001)
Koper, R. 2001 'Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML'. OTEC working paper;
- (Labat, Pernin and Guéraud, 2006)
Labat, J.-M., Pernin, J.-P. and Guéraud, V. 2006 'Contrôle de l'activité de l'apprenant: suivi, guidage pédagogique et scénarios d'apprentissage'. Hermès-Lavoisier.
- (Laborde and Capponi, 1994)
Laborde, C. and Capponi, B. 1994 'Cabri-géomètre constituant d'un milieu pour l'apprentissage de la notion de figure géométrique', *Recherches en didactique des mathématiques*, 14(1.2), pp. 165–210.

(Lando, 2004)

Lando, P. 2004 'Progetto: une méthode de conception de gabarits de scénarios pour activités pédagogiques collectives distantes à base de projets', *Mémoire de DEA. Université de Picardie*.

(Laperrousaz, Leroux and Teutsch, 2007)

Laperrousaz, C., Leroux, P. and Teutsch, P. 2007 'Question de la réutilisation d'outils de suivi d'activités d'apprenants dans des plateformes de formation en ligne', in *Actes de la conférence EIAH 2007*.

(Lefevre *et al.*, 2015)

Lefevre, M., Guin, N., Cablé, B. and Buffa, B. 2015 'ASKER: un outil auteur pour la création d'exercices d'auto-évaluation', in *Atelier EAEI (Évaluation des Apprentissages et Environnements Informatiques)-Conférence EIAH 2015*.

(Leroux, 1993)

Leroux, P. 1993 'A New Development of Control Technology', in Denis, B. (ed.) *Control Technology in Elementary Education*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 241–252. doi: 10.1007/978-3-642-58026-0_19.

(Leroux and Vivet, 2000)

Leroux, P. and Vivet, M. 2000 'Micro-Robots Based Learning Environments for Continued Education in Small and Medium Enterprises (SMEs)', *Journal of Interactive Learning Research*. Association for the Advancement of Computing in Education, 11(3), p. 435.

(De Lièvre, Depover and Dillenbourg, 2006)

De Lièvre, B., Depover, C. and Dillenbourg, P. 2006 'The relationship between tutoring mode and learners' use of help tools in distance education', *Instructional Science*. Springer, 34(2), pp. 97–129.

(Mathenpoche, 2016)

Mathenpoche 2016. Available at: <http://mathenpoche.sesamath.net/>.

(Matheux, 2016)

MatouMatheux 2016. Available at: <http://matoumatheux.ac-rennes.fr/num/proportionnalite/3/fiction.htm>.

(Maxicours, 2016)

Maxicours 2016 *Maxicours*. Available at: <http://www.maxicours.com/>.

(McNamara *et al.*, 2014)

McNamara, D. S., Raine, R., Roscoe, R., Crossley, S. A., Jackson, G. T., Dai, J., Cai, Z., Renner, A., Brandon, R., Weston, J. L. and others 2014 'The Writing-Pal: Natural language algorithms to support intelligent tutoring on writing strategies', in *K-12 Education: Concepts, Methodologies, Tools, and Applications*. IGI Global, pp. 780–793.

(Melis *et al.*, 2001)

Melis, E., Andres, E., Budenbender, J., Frischauf, A., Goduadze, G., Libbrecht, P., Pollet, M. and Ullrich, C. 2001 'ActiveMath: A generic and adaptive web-based learning environment', *International Journal of Artificial Intelligence in Education (IJAIED)*, 12, pp. 385–407.

(Microsoft, 2016)

Microsoft Office 2016. Available at: <https://products.office.com/fr-FR/>.

- (Miled, Laroussi and Reffay, 2015)
Miled, M., Laroussi, M. and Reffay, C. 2015 'An individualised navigation's assistance approach: application and evaluation with the HiPPY prototype', *International Journal of Learning Technology*. Inderscience Publishers (IEL), 10(4), pp. 326–348.
- (Mitrovic, 2003)
Mitrovic, A. 2003 'An intelligent SQL tutor on the web', *International Journal of Artificial Intelligence in Education*. IOS Press, 13(2–4), pp. 173–197.
- (Murray, 1999)
Murray, T. 1999 'Authoring intelligent tutoring systems: An analysis of the state of the art', *International Journal of Artificial Intelligence in Education (IJAIED)*, 10, pp. 98–129.
- (Netbeans, 2014)
Netbeans 2014. Available at: <https://netbeans.org/>.
- (Nogry, Guin and Jean-Daubias, 2008)
Nogry, S., Guin, N. and Jean-Daubias, S. 2008 'AMBRE-add: An ITS to Teach Solving Arithmetic Word Problems.', *Technology, Instruction, Cognition & Learning*, 6(1).
- (Ohlsson, 1986)
Ohlsson, S. 1986 'Some principles of intelligent tutoring', *Instructional science*. Springer, 14(3–4), pp. 293–326.
- (Pachet, 1995)
Pachet, F. 1995 'NéOpus User's Guide'. Laforia: Laforia.
- (Paquette, 1999)
Paquette, G. 1999 'L'ingénierie des interactions dans les systèmes d'apprentissage', *Revue des sciences de l'Éducation*. Revue des sciences de l'éducation, 25(1), pp. 135–161.
- (Paquette, 2012)
Paquette, G. 2012 'Référencement par compétence, recherche et assistance dans les environnements d'apprentissage et de travail'.
- (Paquette, Léonard and Lundgren-Cayrol, 2007)
Paquette, G., Léonard, M. and Lundgren-Cayrol, K. 2007 'The MOT+ visual language for knowledge-based instructional design', *Handbook of Visual Languages for Instructional Design: Theories and Practices: Theories and Practices*. IGI Global.
- (Paquette, Pachet and Giroux, 1994)
Paquette, G., Pachet, F. and Giroux, S. 1994 'ÉpiTalk, un outil générique pour la construction de systèmes conseillers.', *Revue Sciences et Techniques Educatives*, 1(3), pp. 305–336.
- (Paquette *et al.*, 1996)
Paquette, G., Pachet, F., Giroux, S. and Girard, J. 1996 'EpiTalk, a generic tool for the development of advisor systems', *International Journal of Artificial Intelligence in Education*, 7, pp. 349–370.
- (Paquette and Tchounikine, 2002)
Paquette, G. and Tchounikine, P. 2002 'Contribution à l'ingénierie des systèmes conseillers: une approche méthodologique fondée sur l'analyse du modèle de la tâche', *Revue Sciences et Techniques Educatives*, 9(3–4), pp. 157–184.

- (Parisse and Graeve, 2016)
Parisse, B. and Graeve, R. De 2016 *Xcas/Giac*. Available at: https://www-fourier.ujf-grenoble.fr/~parisse/giac_fr.html.
- (Pernin, 2004)
Pernin, J.-P. 2004 'LOM, SCORM et IMS-Learning Design: ressources, activités et scénarios', in *actes du colloque l'indexation des ressources pédagogiques numériques, Lyon*.
- (Pernin and Lejeune, 2004)
Pernin, J.-P. and Lejeune, A. 2004 'Dispositifs d'apprentissage instrumentés par les technologies: vers une ingénierie centrée sur les scénarios', in *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie*, pp. 407–414.
- (Photoscape, 2016)
Photoscape 2016. Available at: <https://photoscape.fr>.
- (Puustinen and Rouet, 2009)
Puustinen, M. and Rouet, J.-F. 2009 'Learning with new technologies: Help seeking and information searching revisited', *Computers & Education*. Elsevier, 53(4), pp. 1014–1019.
- (Quintin, 2007)
Quintin, J.-J. 2007 'L'impact du tutorat sur les performances des étudiants Effet de cinq modalités d'intervention tutorale sur les performances d'étudiants engagés dans un travail collectif asynchrone', in *Actes de la conférence EIAH 2007*.
- (Quintin, Depover and Degache, 2005)
Quintin, J.-J., Depover, C. and Degache, C. 2005 'Le rôle du scénario pédagogique dans l'analyse d'une formation à distance Analyse d'un scénario pédagogique à partir d'éléments de caractérisation définis. Le cas de la formation Galanet'.
- (Richard, 2008)
Richard, B. 2008 Une approche épiphyte pour la conception de systèmes conseillers. Université du Maine.
- (Richterich, 1996)
Richterich, R. 1996 'La compétence stratégique: acquérir des stratégies d'apprentissage et de communication', *Holec, Little & Richterich, op. cit*, pp. 41–76.
- (Ritter and Koedinger, 1995)
Ritter, S. and Koedinger, K. R. 1995 'Towards lightweight tutoring agents', in *AI-ED*, pp. 16–19.
- (Spiral, 2016)
Spiral 2016. Available at: <http://spiralconnect.univ-lyon1.fr/>.
- (TortueLogo, 2016)
Tortue Logo 2016. Available at: <https://www.tortue-logo.fr>.
- (Vygotsky, 1962)
Vygotsky, L. 1962 'Thought and language', *MIT Press*.
- (Winke and MacGregor, 2001)
Winke, P. and MacGregor, D. 2001 'Hot Potatoes version 5', *Language Learning Journal*. Taylor & Francis, 24(1), pp. 30–33.

(Winkels, 1992)

Winkels, R. 1992 Explorations in intelligent tutoring and help. IOS press.

(Yurievich, 2015)

Yurievich, M. A. 2015 'Exercise Maker: Automatic Language Exercise Generation', in Computational Linguistics and Intellectual Technologies. Papers from the Annual International Conference 'Dialogue'(2015), pp. 441–452.

Annexes

Annexe A Logiciels utilisés et étudiés

Logiciels	Pédagogique
ActiveMath (Melis <i>et al.</i> , 2001)	X
ASKER (Lefevre <i>et al.</i> , 2015)	X
MIRTO (Antoniadis <i>et al.</i> , 2005)	X
Exercise Maker (Yurievich, 2015)	X
NetBeans (Netbeans, 2014)	
Ambre-Add (Nogry, Guin and Jean-Daubias, 2008)	X
Tri Sélectif (Jean-Daubias, 2011)	X
Aplusix (Aplusix, 2016)	X
MathEnPoche (Mathenpoche, 2016)	X
IXL Learning (IXL, 2016)	X
Matou Matheux (Matheux, 2016)	X
Educ@ffix (Ergun, 2006)	X
SpiralConnect (Spiral, 2016)	X
Gymglish (Gymglish, 2016)	X
Connectify (Connectify, 2016)	
EOLF (EOLF, 2016)	X
GeoGebra (GeoGebra, 2016)	X
PhotoScape (Photoscape, 2016)	
Gomaths (GoMaths, 2017)	X
Échos d'école (EchosEcole, 2017)	X
Tortue-logo (TortueLogo, 2016)	X
Cabri-géomètre (Laborde and Capponi, 1994)	X
Imagemo (Imagemo, 2017)	X
Maxicours (Maxicours, 2016)	X
MOOC Fun (FunMooc, 2017)	X
Falstad (Falstad, 2017)	X

Figure 164 : Logiciels étudiés pour classifier les caractéristiques de l'assistance en contexte éducatif

Annexe B

Algorithmes utilisés dans les processus

B.1 Algorithmes pour le processus d'application automatique des contraintes des modes d'articulation aux règles

Parcours des blocs racines

```
Si Lbp ne contient pas B
    Ajout des contraintes imposées par le mode d'articulation de B
    Ajout des contraintes imposées par les contraintes sur B
Fin si
```

Ajout des contraintes imposées par le mode d'articulation de B à Lc

```
/*Ajout des contraintes pour la création de la règle d'entrée*/
Nommer le mode d'articulation de B : m
Si m n'est pas successif
    Ajouter la contrainte C_creation_regle(R) à Lc
    Si m est simultané
        Pour tous les blocs, les règles Ri articulées dans B
            Ajouter la contrainte C_alternative(R, Ri, nul) à Lc
        Fin pour
    Fin si

    Si m est progressif
        Pour tous les blocs, les règles Ri articulées dans B avec
un intervalle associé [binfi, bsupi]
            Ajouter la contrainte C_alternative(R, Ri, [binfi,
bsupi]) à Lc
        Fin pour
    Fin si

    Si m est interactif
        Pour tous les blocs, les règles Ri articulées dans B avec
une valeur de comparaison associée ti
            Ajouter la contrainte C_alternative(R, Ri, ti) à Lc
        Fin pour
    Fin si
Fin si

/*Ajout des contraintes aux blocs, règles articulées dans B*/
Si m est successif
    Pour tous les blocs, les règles Ri articulées dans B
        Si i < n
            Ajouter la contrainte C_fin_regle(Ri) à Lc
        Fin si
```

```

        Si i>1
            Ajouter la contrainte C_suite_de_regles(Ri, Ri-1) à Lc
        Fin si
    Fin pour
Sinon
    Pour tous les blocs, les règles Ri articulées dans B
        Ajouter la contrainte C_lancement_regle(Ri) à Lc
    Fin pour
Fin si

```

Ajout des contraintes imposées par le mode d'articulation de B à Lc

```

Pour toutes les contraintes associées Lci à B dans Lc
    Si Lci est la contrainte C_lancement_regle(B)
        Pour le bloc d'entrée ou la règle d'entrée R du bloc B
            Ajouter la contrainte C_lancement_regle(R) à Lc
        Fin pour
    Fin si

    Si Lci est la contrainte C_suite_de_regle(B, X) // (X : n'importe
    quelle règle/ bloc)
        Pour le bloc d'entrée ou la règle d'entrée R du bloc B
            Ajouter la contrainte C_suite_de_regle(R, X) à Lc
        Fin pour
    Fin si

    Si Lci est la contrainte C_debut_regle(B, X) et le mode de B
    n'est pas successif // (X : n'importe quel événement)
        Pour le bloc d'entrée ou la règle d'entrée R du bloc B
            Ajouter la contrainte C_debut_regle(R, X) à Lc
        Fin pour
    Fin si

    Si Lci est la contrainte C_fin_regle(B)
        Pour le bloc d'entrée ou la règle d'entrée R du bloc B
            Ajouter la contrainte C_fin_regle(R) à Lc
        Fin pour
    Fin si
Fin pour

```

Application des contraintes aux règles

```
Pour toutes les contraintes  $L_{ci}$  dans  $L_c$ 
  Si  $L_{ci}$  est la contrainte  $C_{creation\_regle}(X)$  //(X : n'importe
  quelle règle)
    Créer et ajouter une règle X à  $L_r$ 
  Fin si
Fin pour

Pour toutes les règles  $R_i$  dans  $L_r$ 
  Pour toutes les contraintes  $L_{ci}$  associées à  $R_i$  dans  $L_c$  selon un
  ordre de parcours :  $C_{lancement\_regle}$ ,  $C_{suite\_de\_regle}$ ,
 $C_{debut\_regle}$ ,  $C_{nb\_declenchements}$ ,  $C_{consultation}$ ,  $C_{alternative}$ ,
 $C_{fin\_regle}$ 
    Si  $L_{ci}$  est la contrainte est  $C_{lancement\_regle}(R_i)$ 
      Ajouter l'événement déclencheur « lancement par une
  règle » à  $R_i$ .
    Fin si

    Si  $L_{ci}$  est la contrainte est  $C_{suite\_de\_regle}(R_i, X)$  //(X :
  n'importe quelle règle)
      Créer et ajouter l'événement déclencheur « fin de X »
  à  $R_i$ .
    Fin si

    Si  $L_{ci}$  est la contrainte est  $C_{debut\_regle}(R_i, X)$  //(X :
  n'importe quel événement)
      Ajouter l'événement déclencheur X à  $R_i$ .
    Fin si

    Si  $L_{ci}$  est la contrainte est  $C_{nb\_declenchements}(R_i)$ 
      Créer et ajouter la condition « nombre de
  déclenchements de  $R_i$  » à  $R_i$ .
    Fin si

    Si  $L_{ci}$  est la contrainte est  $C_{consultation}(R_i, X)$  //(X :
  n'importe quelle condition)
      Ajouter la condition C à  $R_i$ .
    Fin si

    Si  $L_{ci}$  est la contrainte est  $C_{alternative}(R_i, X, t)$  //(X :
  n'importe quelle alternative, t : n'importe quelle valeur)
      Ajouter X à l'alternative de  $R_i$  (cette alternative
  portant la valeur t est créée est ajoutée à  $R_i$  si  $R_i$  ne la contient
  pas encore)
    Fin si

    Si  $L_{ci}$  est la contrainte est  $C_{fin\_regle}(R_i)$ 
```

```

        Si Ri contient une condition et Ri ne contient pas une
alternative « sinon »
            Ajouter l'alternative « sinon » à Ri
        Fin si

        Si Ri ne contient pas un événement de fin
            Ajouter un message de notification à Lc pour
demande aux concepteurs d'ajouter des événements de fin.
        Fin si
    Fin si

    Fin pour
Fin pour

```

B.2 Algorithmes pour le processus de transformation du guidage pédagogique en règles aLDEAS

```

Créer une règle Rp avec les mêmes événements déclencheurs que le
guidage pédagogique
Pour toutes les activités Ai dans La
    Si Ai ne suit aucune activité précédente
        /* Vérification d'une activité pour Ai */
        Créer une règle Rv avec l'événement déclencheur « lancement
par une règle »
        Créer une condition de type « combinaison » C qui combine
toutes les conditions pour Ai *
        Ajouter C à Rv
        Ajouter l'action « ajouter Ai dans la liste d'activités »
à l'alternative de Ry qui porte une valeur « vrai »

        /* lancement d'une activité pour Ai */
        Créer une règle Rl avec l'événement déclencheur « choix de
l'activité Ai par l'utilisateur »
        Ajouter l'action « lancer l'activité Ai » à Rl

        /* fin d'une activité pour Ai */
        Ajouter l'événement déclencheur « fin de Ai » dans Rp
        Ajouter la règle Rv à Rp
    Fin si
Ajouter l'action « afficher la liste d'activités » dans Rp

```

* Opérateur logique « && » (respectivement « || ») entre les conditions et les états de sortie des activités précédentes pour la condition « et » (respectivement « ou »).

Annexe C

Typologie des éléments aLDEAS

Les nouvelles actions d'assistance élémentaires et les nouveaux éléments élémentaires issus de cette thèse nécessitent l'évolution de la typologie des éléments aLDEAS. La Figure 165 donne la typologie des événements aLDEAS et la Figure 166 donne la typologie des actions d'assistance élémentaires aLDEAS. Les nouveaux éléments issus de cette thèse sont notés en vert dans les figures.

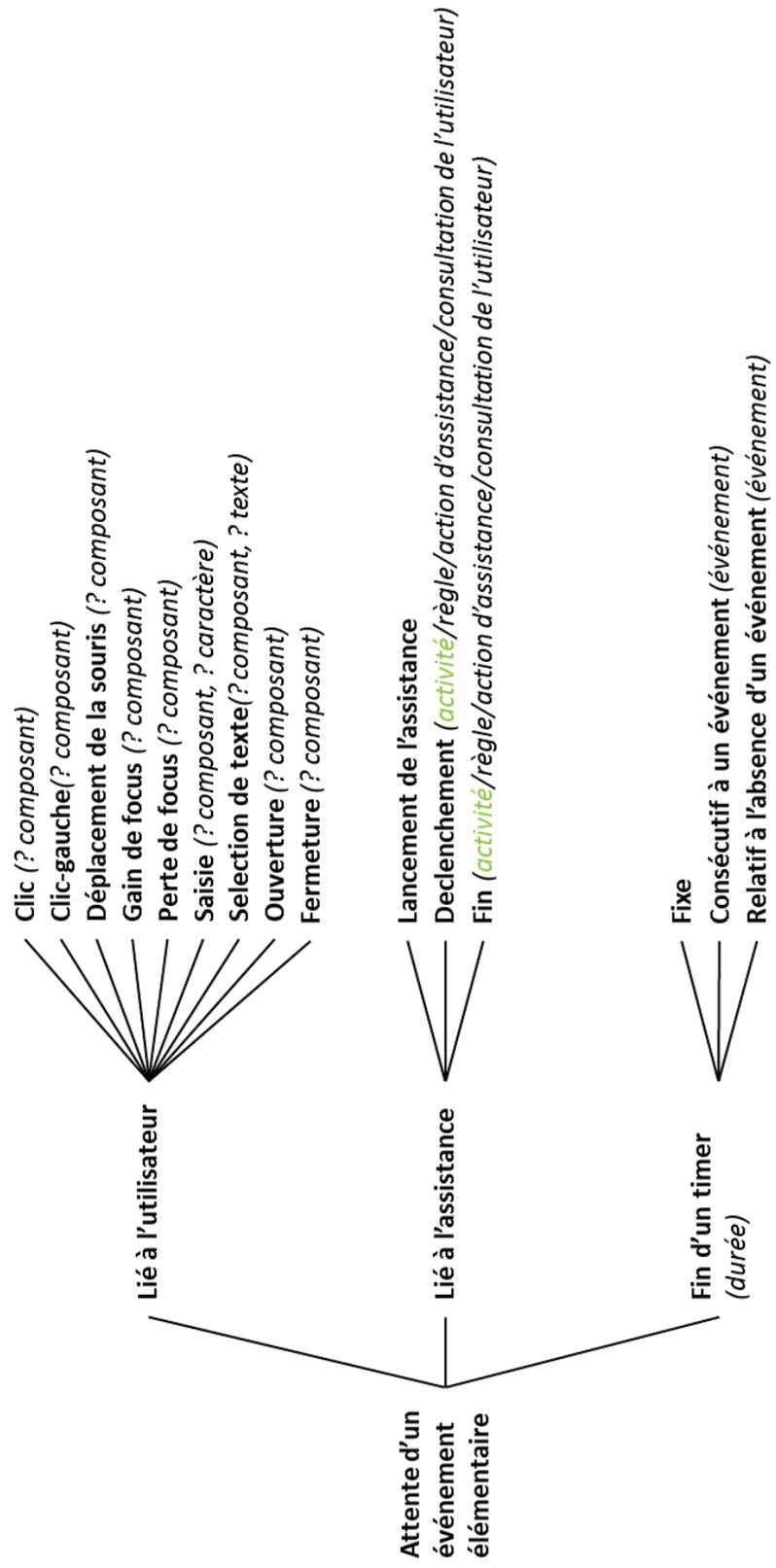


Figure 165 : Typologie des événements élémentaires aLDEAS

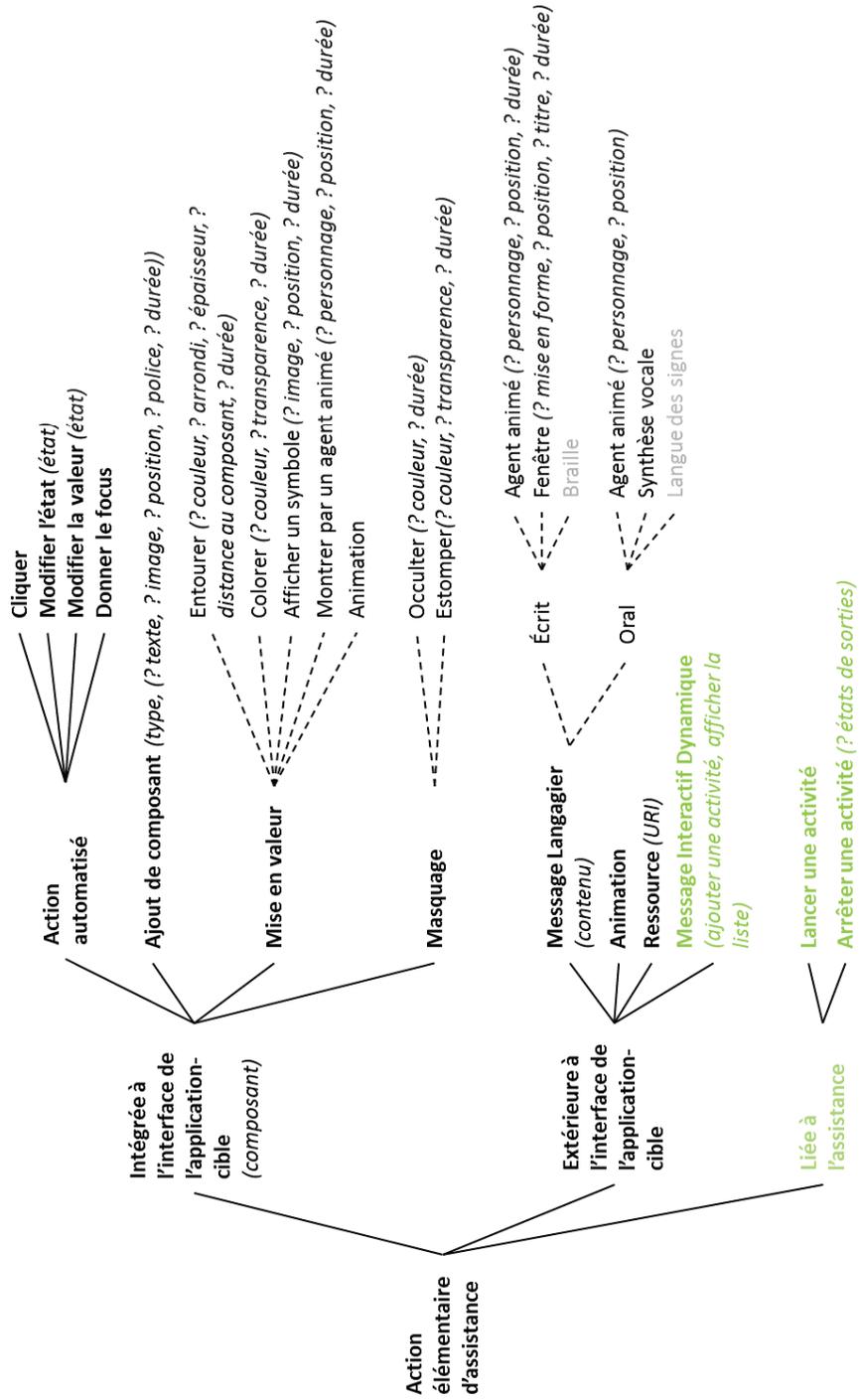


Figure 166 : Typologie des actions d'assistance élémentaires aLDEAS

Annexe D

Exemples d'assistance définies et exécutées avec SEPIA-edu

Pour valider nos contributions, nous avons défini un certain nombre de systèmes d'assistance avec SEPIA-edu sur les applications-cibles en contexte éducatif. Nous montrons ici des captures d'écran qui illustrent la définition de tels systèmes d'assistance avec l'éditeur de SEPIA-edu, elles sont complétées par des vidéos de démonstration visibles sur le site du projet AGATE (AGATE, 2016).

D.1 Exemples de déroulement d'une assistance

Exemple sur NetBeans

Le système d'assistance permettant d'afficher un tutoriel sur NetBeans (scénario d'usage 3 de Chapitre 1) a été défini et exécuté avec SEPIA-edu. La Figure 167 illustre la définition du tutoriel par deux blocs de règles articulées en mode successif.

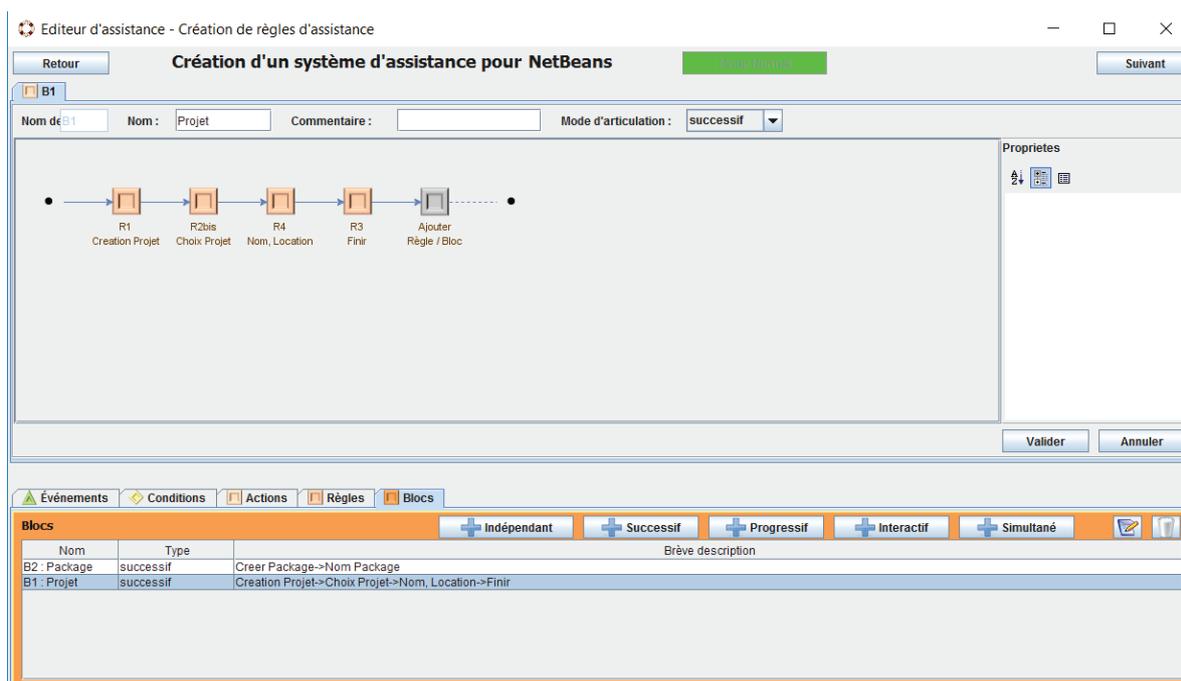


Figure 167 : Écran SEPIA-edu de définition du tutoriel sur NetBeans

Exemple sur le site GeoGebra

Le système d'assistance permettant d'afficher les instructions explicitant l'activité sur GeoGebra (scénario d'usage 2 de Chapitre 1) a été défini et exécuté avec SEPIA-edu. La Figure 168 illustre la définition de ce système par un bloc de règles articulées en mode successif.

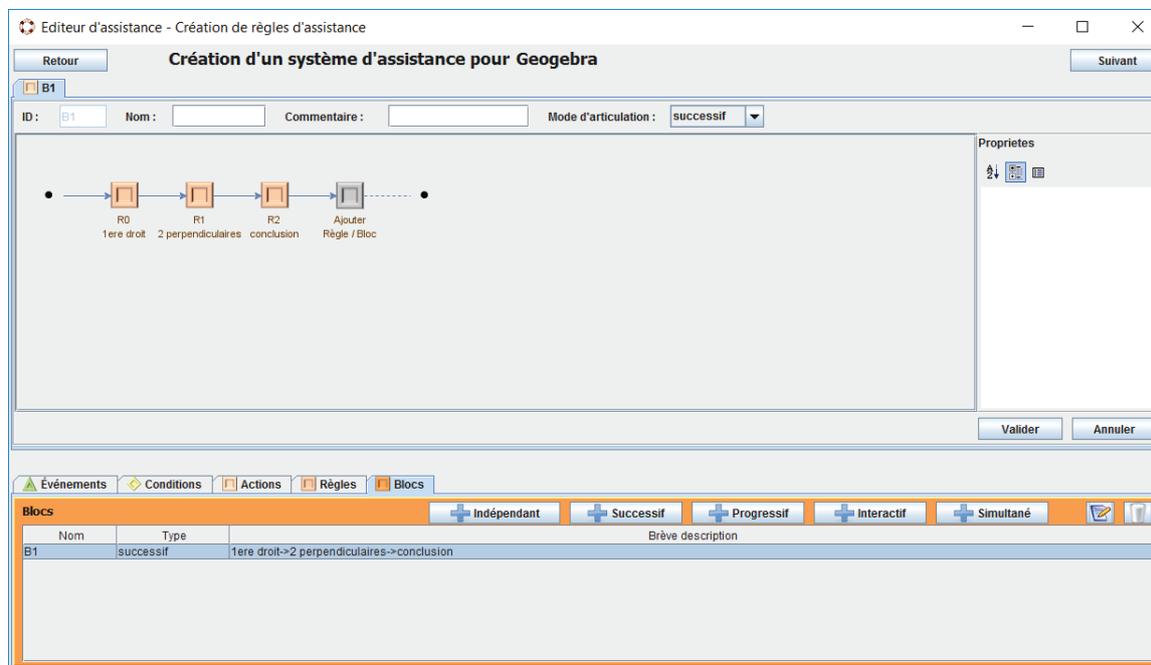


Figure 168 : Écran SEPIA-edu de définition des instructions explicitant l'activité sur GeoGebra avec SEPIA-edu

Exemple sur le site IXL Learning

Le système d'assistance permettant d'afficher la vue d'ensemble de chaque cours sur IXL Learning (exemple présenté dans la section 4.1.2) a été défini et exécuté avec SEPIA-edu. La Figure 169 illustre la définition du tutoriel par un bloc de trois règles articulées en mode successif.

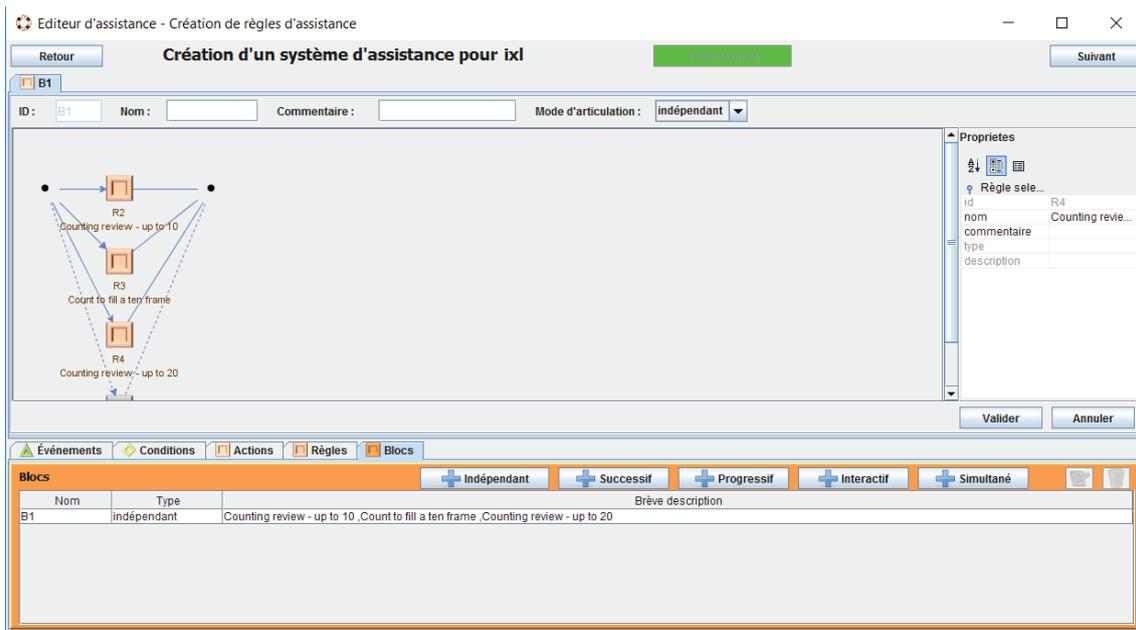


Figure 169 : Écran SEPIA-edu de définition de l'assistance sur IXL Learning

Exemple sur le site Matou Matheux

Sur le site Matou Matheux, nous avons défini cinq systèmes d'assistance correspondant à nos cinq modes d'articulation (systèmes d'assistance utilisés pour l'expérimentation de 2015) et un système d'assistance combinant les deux modes d'articulation progressif et successif (scénario d'usage 1 de Chapitre 1). La Figure 170 illustre la définition de ce dernier. Avec SEPIA-edu, ce système est défini par deux blocs combinés de règles articulées, en mode progressif pour l'un et en mode successif pour l'autre.

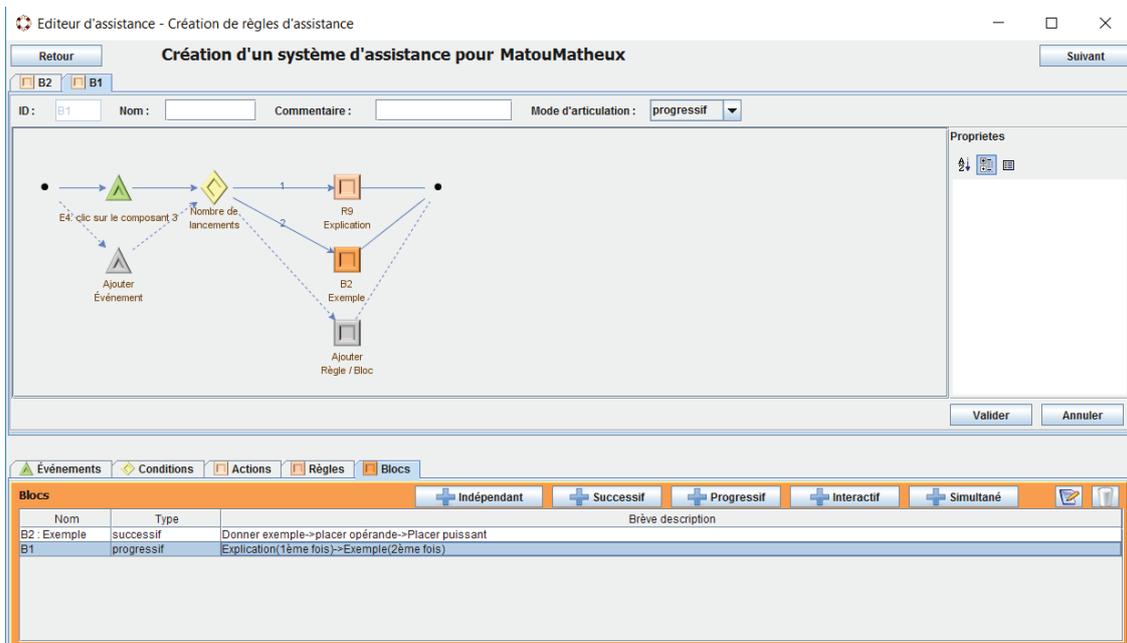


Figure 170 : Écran SEPIA-edu de définition de l'assistance sur Matou Matheux

Exemple sur l'application Connectify

Le système d'assistance permettant d'afficher un tutoriel (exemple présenté dans la section 4.1.2) a été défini et exécuté avec SEPIA-edu. La Figure 171 illustre la définition du tutoriel par un bloc de trois règles articulées en mode successif.

Nom	Evènement déclencheur	Condition	Action	événement de fin
R1	Lancement de l'assistance		A0	
R0 : Bienvenue	E2 : mouseClicked sur le composant C0		A1	E3 : 10 secondes
R2 : Vérification	EG0 : fin sur le composant R0		A2	E3 : 10 secondes
R3 : Choix	EG1 : fin sur le composant R2		A3, A4	E3 : 10 secondes

Figure 171 : Écran SEPIA-edu de définition du tutoriel sur Connectify

Exemple sur le site EOLF

Le système d'assistance permettant de faire le diagnostic sur le site EOLF (exemple présenté dans la section 4.1.3) a été défini et exécuté avec SEPIA-edu. La Figure 172 illustre la définition du diagnostic parallèle s'appuyant sur trois questions par un bloc de trois règles articulées en mode simultané.

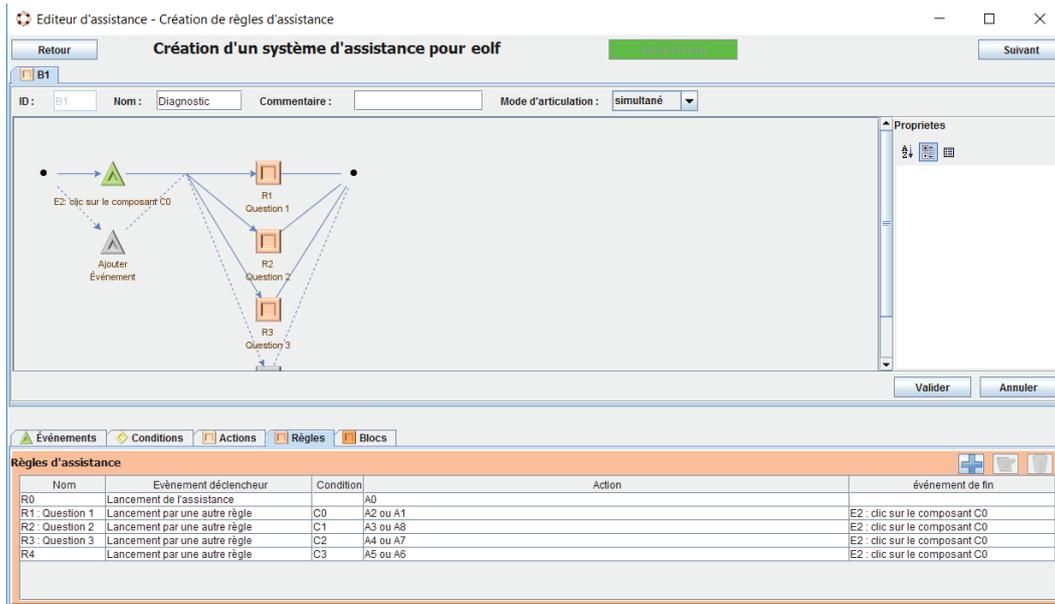


Figure 172 : Écran SEPIA-edu de définition du diagnostic sur EOLF

Exemple sur ActiveMath

Le système d'assistance permettant de fournir les indices, puis la solution de façon progressive sur le site ActiveMath (exemple présenté dans la section 4.1.4) a été défini et exécuté avec SEPIA-edu. La Figure 173 illustre la définition du diagnostic parallèle de trois questions par un bloc de trois règles articulées en mode progressif. Cependant, le lien de ce site est brisé. Nous ne pouvons donc plus exécuter le système d'assistance sur ce site pour réaliser une vidéo de démonstration.

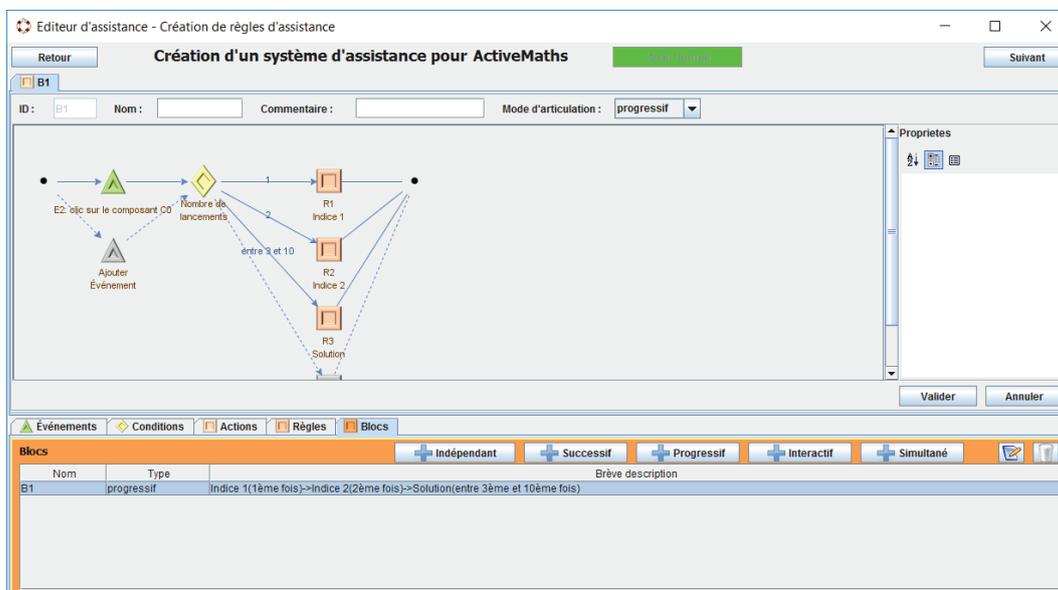


Figure 173 : Écran SEPIA-edu de définition de l'assistance progressive sur ActiveMath

D.2 Exemples de guidage pédagogique

Exemple sur un MOOC Fun

Un guidage pédagogique proposé sur le MOOC Fun « Racines » a été défini et exécuté avec SEPIA-edu. Ce guidage se compose de trois activités qui sont proposées les unes après les autres. La Figure 174 illustre la définition du guidage pédagogique. La condition C₅ (respectivement C₂₀) assure que l'activité 1 (respectivement l'activité 2) n'est plus proposée si l'apprenant l'a réussie. En cas d'échec à l'activité 2, l'activité 1 est proposée de nouveau à l'apprenant. Ce guidage a également été exécuté sur le MOOC. La vidéo de démonstration disponible sur le site du projet AGATE illustre cette exécution.

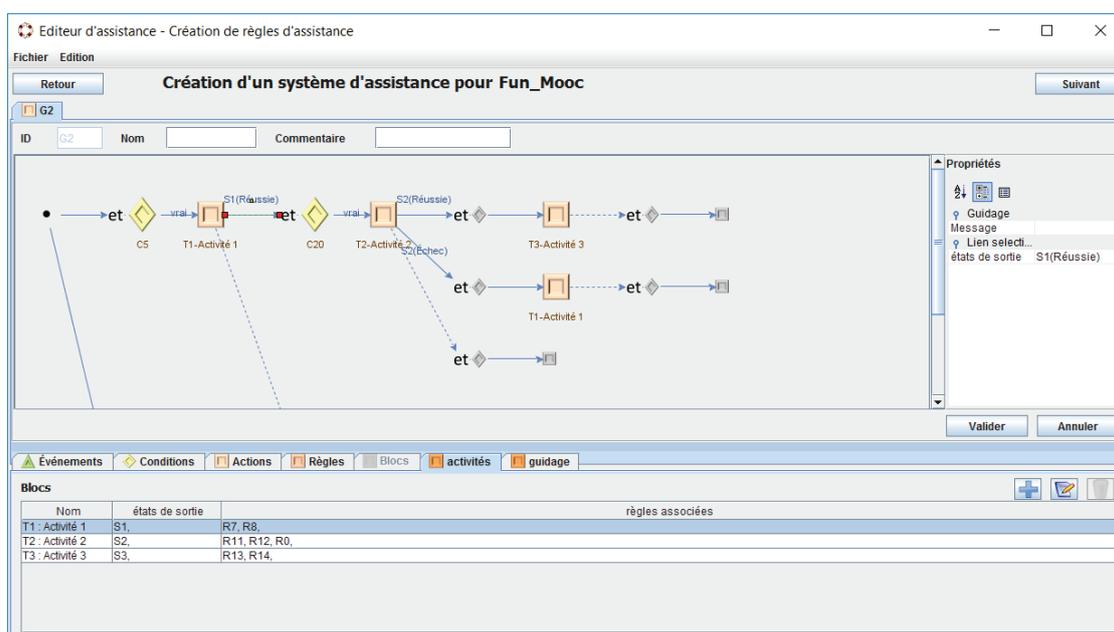


Figure 174 : Écran SEPIA-edu de définition d'un guidage pédagogique sur le MOOC Fun « Racines »

Exemple sur ASKER

Un guidage pédagogique proposé sur le site ASKER a été défini et exécuté avec SEPIA-edu. Ce guidage se compose de 13 activités. La Figure 175 illustre la définition de ce guidage. Les activités sont proposées les unes après les autres. En particulier, certaines activités sont proposées seulement si plusieurs activités sont réalisées. Par exemple, l'activité 125 est proposée si les activités 117, 118 et 119 sont réalisées.

Ce guidage pédagogique a été exécuté de manière opérationnelle avec SEPIA-edu. Toutefois, des changements intervenus sur le site qui héberge ASKER ne nous permet plus de réaliser de vidéo de démonstration.

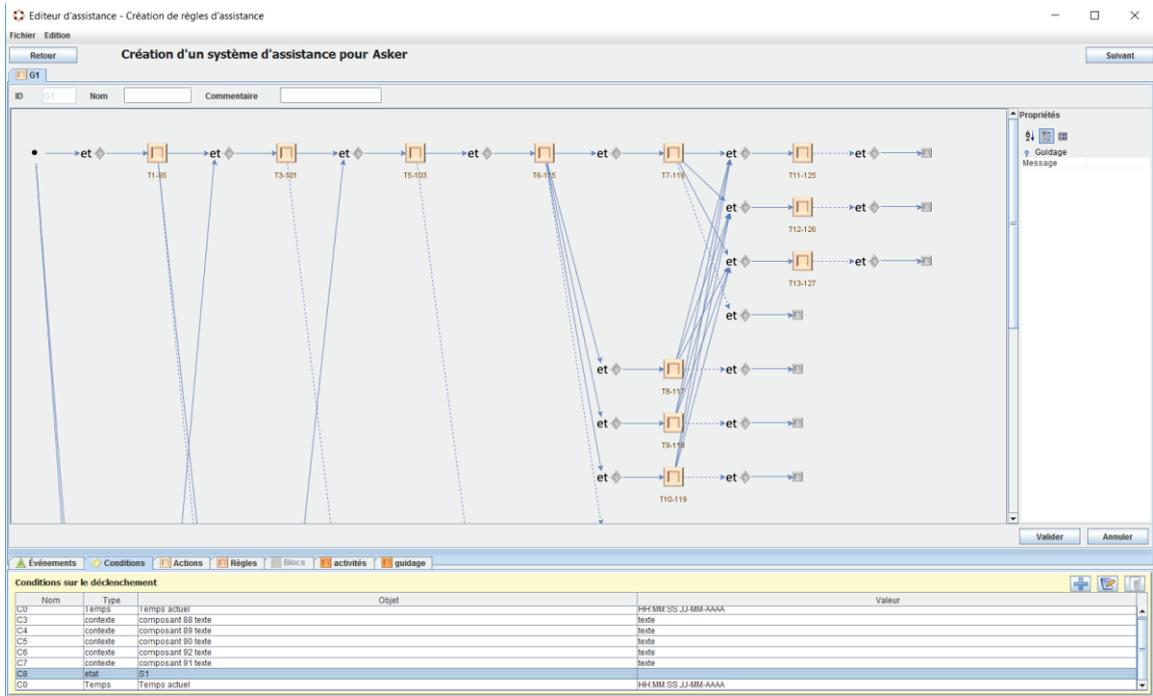


Figure 175 : Écran SEPIA-edu de définition d'un guidage pédagogique sur ASKER

Annexe E

Documents pour les mise à l'essai

E.1 Mise à l'essai en 2015

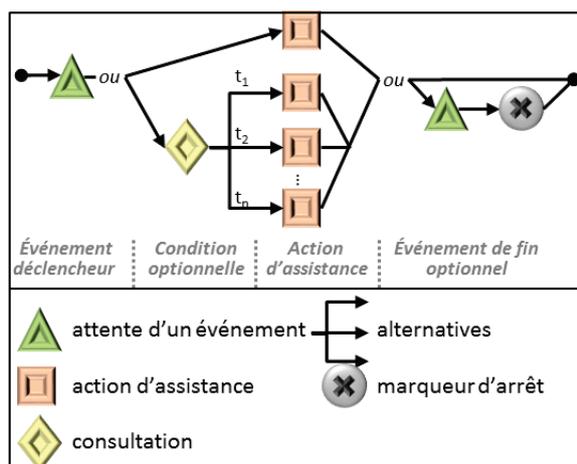
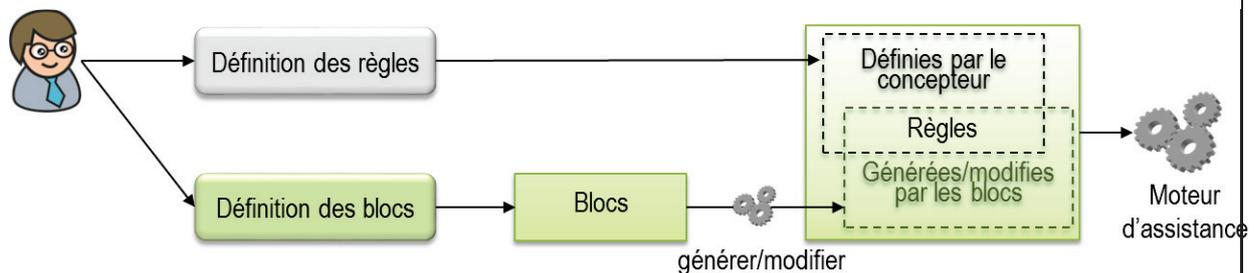
Nous avons préparé cinq documents correspondants aux cinq modes d'articulation. Ces documents comportent une introduction de SEPIA-edu, les étapes à suivre par les sujets et un questionnaire à compléter. En raison de la longueur de ces documents, nous ne présentons ci-dessous que l'exemple du document correspondant au mode d'articulation successif.

Assistance à l'utilisateur : utilisation de SEPIA pour consulter, modifier et définir des systèmes d'assistance

Dans ce TP, vous allez ajouter un système d'assistance à certains exercices de Matou Matheux. Ce site Web de l'académie de Rennes propose un grand nombre de ressources pédagogiques mises librement à disposition des enseignants. Pour créer ce système d'assistance, vous allez utiliser SEPIA, le logiciel issu du projet AGATE.

Le projet **AGATE** vise à étudier comment faciliter l'utilisation de logiciels par l'adjonction de systèmes d'assistance épiphytes (c'est-à-dire des systèmes d'assistance qui se greffent sur un logiciel existant, nommé application-cible, sans en perturber le fonctionnement). Le système **SEPIA** opérationnalise cette idée. Il se compose d'un éditeur d'assistance et d'un moteur d'assistance. L'**éditeur d'assistance** est l'outil dédié aux concepteurs d'assistance pour créer des systèmes d'assistance. Les systèmes d'assistance créés sont ensuite exécutés par le **moteur d'assistance** pour fournir de l'assistance lors de l'exécution de l'application-cible pour les utilisateurs finaux. Ouvrez le fichier Utilisation_Generale.mp4 pour voir la démonstration de l'utilisation générale de SEPIA.

La communication entre l'éditeur et le moteur d'assistance est rendue possible par un langage pivot, le **langage aLDEAS**. Ainsi, un système d'assistance est défini par un ensemble de règles aLDEAS et leur enchaînement. Il est possible de regrouper des règles sous forme d'un **bloc** associé à un mode d'articulation entre ces règles.



Une **règle** commence par un **événement déclencheur** (une action de l'utilisateur, un événement lié à l'assistance, une durée). Dès que cet événement a lieu, les **actions d'assistance** qui suivent (blocs de règles, règles, actions élémentaires telles que message, mise en valeur, ajout d'un bouton d'aide...) sont lancées soit immédiatement, soit en fonction d'une **condition**. Une condition prend la forme

d'une **consultation** (de l'état de l'application, de l'historique de l'assistance...) avec des alternatives associées chacune à une action. Enfin, la règle peut se terminer par un **événement de fin** qui met fin à toutes les actions élémentaires déclenchées par cette règle.

Cinq modes d'articulation entre règles sont disponibles pour définir des blocs :

- Mode **indépendant** : les règles ne sont a priori pas liées les unes aux autres, elles sont lancées indépendamment par leurs propres événements déclencheurs,
- Mode **successif** : les règles sont lancées les unes après les autres,
- Mode **simultané** : toutes les règles du bloc sont lancées au même moment,
- Mode **progressif** : les règles sont lancées au fur et à mesure des besoins d'assistance, permettant une progression du degré d'intervention du système,
- Mode **interactif** : le lancement des règles est contraint par une condition (question posée à l'utilisateur, état de son profil, etc.).

Le système d'assistance que vous allez contribuer à mettre en place vise à adapter des exercices de Matou Matheux pour qu'ils soient en adéquation avec les objectifs donnés d'un enseignant donné pour un contexte donné (adapté à un niveau scolaire, aux élèves, etc.).

L'exercice à travailler concerne « les durées et les vitesses »

<http://matoumatheux.ac-rennes.fr/num/proportionnalite/3/fiction.htm>

Dans cet exercice, un élève de niveau 3^{ème} doit convertir une vitesse de km/s en km/h. L'élève doit ensuite écrire le résultat dans une écriture scientifique (sous la forme $a \times 10^b$, en saisissant les valeurs de a et b dans les deux champs correspondants, a = opérande et b = puissance de 10).

Ce TP est organisé en quatre parties d'utilisation progressive de l'outil SEPIA pour créer un système d'assistance :

1. compréhension de la modélisation d'un système d'assistance existant,
2. modification de ce système d'assistance,
3. complétion de ce système d'assistance,
4. création d'un nouveau système d'assistance.

Dans ce TP, vous allez travailler avec l'éditeur d'assistance de SEPIA pour **comprendre, améliorer** et **créer** un système d'assistance de 3 manières différentes faisant appel à 3 types de représentations T1 T2 T3 :

- T1. **Avec des règles, sans bloc**, sous forme de représentation **textuelle** : vous utiliserez seulement les règles pour définir un système d'assistance en utilisant l'interface textuelle de définition des règles,
- T2. **Avec des règles, sans bloc**, sous forme de représentation **graphique** : vous utiliserez seulement les règles pour définir un système d'assistance en utilisant l'interface graphique de définition des règles,
- T3. **Avec des blocs** : vous utiliserez les règles et les blocs pour définir un système d'assistance.

Type de représentation 1 : règles sans bloc avec interface textuelle

Dans cette partie vous allez utiliser SEPIA en mode « interface textuelle ».

Ouvrez le fichier Interface_Textuelle.mp4 pour voir la démonstration de l'utilisation de l'interface textuelle de SEPIA. Le mode « interface textuelle » est choisie par défaut.

T1 – étape 1 : compréhension d'un système d'assistance existant

Dans l'éditeur d'assistance de SEPIA, sélectionnez le logiciel Matou Matheux, puis chargez le fichier MatouMatheux_1.xml : il s'agit de la modélisation d'un système d'assistance pour présenter **des niveaux scolaires abordés** par Matou Matheux.

Étudiez le système d'assistance de cet exercice en examinant les **règles** dans l'éditeur d'assistance : laquelle des assistances décrites ci-dessous correspond à celle proposée (entourez le n° correspondant à votre choix ①②③④⑤) ?

- ① Au lancement du système d'assistance, les messages et les mises en valeur sont affichés les uns après les autres pour guider un élève à choisir des cours adaptés à son niveau scolaire. Il y a trois étapes d'assistance successives :
 - un premier message de bienvenue
 - un message et une mise en valeur présentent la liste des cours pour le niveau CP-CE1.
 - un message et une mise en valeur présentent la liste des cours pour le niveau CE2-CM1.
- ② Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, les messages et les mises en valeur sont affichés les uns après les autres pour le guider à choisir des cours pertinents à son niveau d'apprentissage. Il y a trois étapes d'assistance successives :
 - un premier message de bienvenue
 - un message et une mise en valeur présentent la liste des cours pour le niveau CP-CE1.
 - un message et une mise en valeur présentent la liste des cours pour le niveau CE2-CM1.
- ③ Au lancement, le système d'assistance affiche 3 étapes de présentation des cours en même temps :
 - un premier message de bienvenue
 - un message et une mise en valeur présentent la liste des cours pour le niveau CE2-CM1.
 - un message et une mise en valeur présentent la liste des cours pour le niveau CP-CE1.
- ④ Au lancement du système d'assistance, les messages et les mises en valeur sont affichés les uns après les autres pour guider un élève à choisir des cours pertinents à son niveau d'apprentissage. Il y a trois étapes d'assistance successives :
 - un premier message de bienvenue
 - un message et une mise en valeur présentent la liste des cours pour le niveau CE2-CM1.
 - un message et une mise en valeur présentent la liste des cours pour le niveau CP-CE1.

⑤ Autre, décrivez le déroulement de l'assistance :

T1 – étape 2 : modification d'un système d'assistance existant

Voici, la description du système d'assistance que vous venez d'examiner :

Au lancement du système d'assistance, les messages et les mises en valeur sont affichés les uns après les autres pour guider l'élève dans le choix des cours adaptés à son niveau d'apprentissage. Il y a trois étapes d'assistance successives :

- un premier message de bienvenue
- un message et une mise en valeur présentent la liste des cours pour le niveau CE2-CM1
- un message et une mise en valeur présentent la liste des cours pour le niveau CP-CE1

Dans cette assistance, il serait plus logique de présenter les cours du niveau CP-CE1 avant ceux du niveau CE2-CM1. Pour améliorer le système d'assistance, intervertissez les étapes 2 et 3.

Pour cela, toujours à l'aide de la version « règle avec présentation textuelle » de SEPIA, modifiez ce système d'assistance. Validez les modifications, puis testez la nouvelle version de l'assistance.

Après avoir modifié ce système d'assistance, répondez aux questions suivantes :

Avez-vous réussi à effectuer cette interversion dans le système d'assistance ? oui non

Combien d'essais avez-vous fait avant de réussir ?

T1 – étape 3 : complétion d'un système d'assistance existant

Dans ce même système d'assistance de cet exercice de Matou Matheux, il serait pertinent d'ajouter une étape d'assistance qui présente le niveau CM2 qui a été oublié dans la première version de l'assistance :

- un **message** «En cliquant sur CM2, tu peux suivre les cours adaptés à cette classe : nombres entiers, décimaux, division, fractions, pourcentages, proportionnalité, etc. »
- une **mise en valeur** de l'item de menu « CM2 »
- tous deux affichés **pendant 5 secondes**.

Pour cela, toujours à l'aide de la version « règle avec présentation textuelle » de SEPIA, complétez ce système d'assistance. Validez les modifications, puis testez la nouvelle version de l'assistance.

Après avoir complété ce système d'assistance, répondez aux questions suivantes :

Avez-vous réussi à compléter le système d'assistance ? oui non

Combien d'essais avez-vous fait avant de réussir ?

Type de représentation 2 : règles sans bloc avec présentation graphique

Dans cette partie vous allez utiliser SEPIA en mode « interface graphique ».

Ouvrez le fichier Interface_Graphique .mp4 pour voir la démonstration de l'utilisation de l'interface graphique de SEPIA. Le mode « interface graphique » est choisie par défaut.

T2 – étape 1 : compréhension d'un système d'assistance existant

Chargez maintenant le fichier MatouMatheux_2.xml dans l'éditeur d'assistance de SEPIA : il s'agit de la modélisation d'un système d'assistance pour présenter [l'organisation de la page de Web](#) de Matou Matheux correspondant à l'exercice « les durées et les vitesses ».

Étudiez le système d'assistance de cet exercice en examinant les **règles** dans **l'éditeur d'assistance** : laquelle des assistances décrites ci-dessous correspond à celle proposée (entourez le n° correspondant à votre choix ①②③④⑤) ?

- ① Au lancement du système d'assistance, les messages d'explication et les mises en valeur sont affichés les uns après les autres. Il y a trois étapes d'assistance successives :
 - Un message explique l'organisation globale de la page Web.
 - Un message et une flèche présentent la zone de travail de l'utilisateur.
 - Un message et une flèche entourage présentent le menu de cette page.
- ② Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, les messages d'explication et les mises en valeur sont affichés les uns après les autres selon trois étapes d'assistance successives :
 - Un message explique l'organisation globale de la page Web.
 - Un message et une flèche présentent la zone de travail de l'utilisateur.
 - Un message et une flèche présentent le menu de cette page.
- ③ Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, un message expliquant l'organisation globale de la page Web. Quand l'utilisateur clique une deuxième fois sur le bouton d'aide, un message et une flèche expliquant la zone de travail de l'utilisateur. Quand l'utilisateur clique troisième fois sur le bouton d'aide, un message et une flèche expliquant la zone de travail de l'utilisateur.
- ④ Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, le système d'assistance affiche trois étapes de présentation de la page Web en même temps :
 - un message explique l'organisation globale de cette page.
 - un message et une flèche présentent la zone de travail de l'utilisateur.
 - un message et une flèche présentent le menu de cette page
- ⑤ Autre, décrivez :

T2 – étape 2 : modification d'un système d'assistance existant

Voici, la description du système d'assistance que vous venez d'examiner :

Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, les messages d'explication et les mises en valeur sont affichés les uns après les autres selon trois étapes d'assistance successives :

- Un message explique l'organisation globale de la page Web.
- Un message et une flèche présentent la zone de travail de l'utilisateur.
- Un message et une flèche présentent le menu de cette page.

Dans cette page, il serait plus logique de présenter le menu avant la zone de travail de l'utilisateur. Pour améliorer le système d'assistance, intervertissez les deux étapes successives.

Pour cela, toujours à l'aide de la version « règle avec présentation textuelle » de SEPIA, modifiez ce système d'assistance. Validez les modifications, puis testez la nouvelle version de l'assistance.

Après avoir modifié ce système d'assistance, répondez aux questions suivantes :

Avez-vous réussi à modifier le système d'assistance ? oui non

Combien d'essais avez-vous fait avant de réussir ?

T2 – étape 3 : complétion d'un système d'assistance existant

Dans le système d'assistance de cette page, il serait pertinent d'ajouter une étape d'assistance : un **message** « Pour terminer, tu cliques sur le bouton « ? » après d'avoir répondu à la question » et une **flèche** pointant sur le bouton « ? » pendant **5 secondes**.

Pour cela, toujours à l'aide de la version « règle avec présentation graphique » de SEPIA, complétez ce système d'assistance. Validez les modifications, puis testez la nouvelle version de l'assistance.

Après avoir complété ce système d'assistance, répondez aux questions suivantes :

Avez-vous réussi à compléter le système d'assistance ? oui non

Combien d'essais avez-vous fait avant de réussir ?

Type de représentation 3 : blocs

Dans cette partie vous allez utiliser SEPIA en mode « interface graphique » avec blocs.

Ouvrez le fichier Bloc .mp4 pour voir la démonstration de l'utilisation en mode « interface graphique » de SEPIA avec le bloc. Le mode « interface graphique » est choisie par défaut.

T3 – étape 1 : Compréhension d'un système d'assistance existant

Dans l'éditeur d'assistance de SEPIA, sélectionnez le logiciel Matou Matheux, puis chargez le fichier MatouMatheux_3.xml : il s'agit de la modélisation d'un système d'assistance pour l'exercice « [Les durées et les vitesses](#) » de Matou Matheux.

Étudiez le système d'assistance de cet exercice en examinant les **règles** ainsi que le **bloc** qui montre l'articulation entre les règles dans **l'éditeur d'assistance** : laquelle des assistances décrites ci-dessous correspond à celle proposée (entourez le n° correspondant à votre choix ①②③④⑤) ?

- ① Au lancement du système d'assistance, les messages d'explication et les mises en valeur sont affichés les uns après les autres. Il y a deux étapes d'assistance successives :
- Un message demande de convertir la vitesse km/s en km/h.
 - Un message et une mise en valeur demandent de saisir la valeur du champ « exposant ».
 - Un message et une mise en valeur demandent de saisir la valeur du champ « opérande ».
- ② Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, les messages d'explication et les mises en valeur sont affichés les uns après les autres selon deux étapes d'assistance successives :
- Un message demande de convertir la vitesse km/s en km/h.
 - Un message et une mise en valeur demandent de saisir la valeur du champ « exposant ».
 - Un message et une mise en valeur demandent de saisir la valeur du champ « opérande ».
- ③ Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, un message demandant de convertir la vitesse km/s en km/h . Quand l'utilisateur clique deuxième fois sur le bouton d'aide, un message et une mise en valeur demandant de saisir la valeur du champ « exposant » sont affichés. Quand l'utilisateur clique troisième fois sur le bouton d'aide, un message et une mise en valeur demandant de saisir la valeur du champ « opérande » sont affichés.
- ④ Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, un message demandant de convertir la vitesse km/s en km/h sont affichés en même temps que 2 messages et 2 mises en valeur demandant de saisir la valeur du champ « exposant » et « opérande ».
- ⑤ Autre, décrivez :

T3 – étape 2 : modification d'un système d'assistance existant

Voici, la description du système d'assistance que vous venez d'examiner :

Au lancement, le système d'assistance affiche un bouton d'aide. Quand l'utilisateur clique sur ce bouton, les messages d'explication et les mises en valeur sont affichés les uns après les autres selon deux étapes d'assistance successives :

- Un message demande de convertir la vitesse km/s en km/h
- Un message et une mise en valeur demandent de saisir la valeur du champ « exposant »
- Un message et une mise en valeur demandent de saisir la valeur du champ « opérande »

Dans cet exercice de Matou Matheux, il serait plus logique de demander la saisie de l'opérande avant celle de l'exposant. Pour améliorer le système d'assistance, intervertissez les deux étapes successives.

Pour cela, toujours à l'aide de la version « bloc » de SEPIA, modifiez ce système d'assistance. Validez les modifications, puis testez la nouvelle version de l'assistance.

Après avoir modifié ce système d'assistance, répondez aux questions suivantes :

Avez-vous réussi à modifier le système d'assistance ? oui non

Combien d'essais avez-vous fait avant de réussir ?

T3 – étape 3 : complétion d'un système d'assistance existant

Dans le système d'assistance de cet exercice de Matou Matheux, il serait pertinent d'ajouter une étape d'assistance après la première étape : un **message** « Ensuite, tu dois transformer la vitesse au format $a \cdot 10^b$ » affiché **pendant 5 secondes** qui demande à l'utilisateur de transformer la vitesse au format $a \cdot 10^b$.

Pour cela, toujours à l'aide de la version « bloc » de SEPIA, complétez ce système d'assistance. Validez les modifications, puis testez la nouvelle version de l'assistance.

Après avoir complété ce système d'assistance, répondez aux questions suivantes :

Avez-vous réussi à compléter le système d'assistance ? oui non

Combien d'essais avez-vous fait avant de réussir ?

Conclusion (étape 4)

Vous allez créer un système d'assistance suivant :

Au lancement, le système d'assistance affiche un bouton d'aide à côté du bouton « ? ». Quand l'utilisateur clique sur ce bouton, un exemple est fourni à l'utilisateur sous forme de 3 étapes d'assistance successives :

- Un **message** affiche tout d'abord « Dans notre exemple, la vitesse est de 10 km/s. Il faut commencer par convertir cette vitesse en km/h : $10 \times 3600 \text{ km/h} = 36000 \text{ km/h} = 3,6 \cdot 10^4 \text{ km/h}$ » pendant **5 secondes** ;
- Un deuxième **message** affiche ensuite « Il faut mettre dans ce champ la partie entière de la valeur convertie, ici 3,6 » pendant que le champ *opérande* est **mis en valeur**. Ces message et mise en valeur existent pendant **5 secondes** ;
- Un **message** affiche enfin « Il faut ensuite mettre dans ce champ la puissance de 10 de la valeur convertie, ici 4 » pendant que le champ *exposant* est **mis en valeur**. Ces message et mise en valeur existent pendant **5 secondes**.

Si vous deviez utiliser SEPIA pour créer ce système d'assistance, quelle version de SEPIA préféreriez-vous utiliser ?

- version règle représentation textuelle (MatouMatheux _1.xml)
- version règle représentation graphique (MatouMatheux _2.xml)
- version bloc (MatouMatheux _3.xml)

À l'aide de votre version préférée de SEPIA, créez ce système d'assistance nommé MatouMatheux_4.xml. Validez les modifications, puis testez la nouvelle version de l'assistance.

Après l'avoir créé, répondez aux questions suivantes :

Avez-vous réussi à créer le système d'assistance ? oui non

Estimez la facilité de compréhension d'un système d'assistance, selon les représentations du système d'assistance, laquelle préférez-vous (entourez) ?

version règle représentation textuelle	réussi	très difficile	difficile	neutre	facile	très facile	préférée
version règle représentation graphique	réussi	très difficile	difficile	neutre	facile	très facile	préférée
version bloc	réussi	très difficile	difficile	neutre	facile	très facile	préférée

Décrivez les raisons :

Estimez la facilité de définition d'un système d'assistance, selon les représentations du système d'assistance, laquelle préférez-vous (entourez) ?

version règle représentation textuelle	réussi	très difficile	difficile	neutre	facile	très facile	préférée
version règle représentation graphique	réussi	très difficile	difficile	neutre	facile	très facile	préférée
version bloc	réussi	très difficile	difficile	neutre	facile	très facile	préférée

Décrivez les raisons :

Dans la version bloc de SEPIA il y a actuellement cinq modes d'articulation entre les règles : indépendant, successif, simultané, interactif, progressif. Avez-vous déjà rencontré dans votre expérience d'assistance humaine ou informatique un autre mode ?

oui non

Si oui, décrivez ce mode d'articulation :

Si non, décrivez les raisons :

E.2 Mise à l'essai en 2017

Nous avons préparé un document unique pour tous les étudiants. Ce document comporte une introduction de SEPIA-edu, les étapes à suivre par les sujets et un questionnaire à compléter. Nous le présentons ci-dessous.

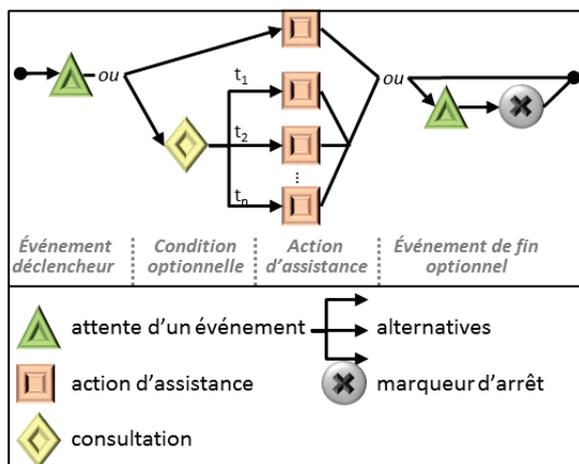
Systèmes d'assistance à l'utilisateur épiphytes avec SEPIA

Dans ce TP, vous allez ajouter un système d'assistance au MOOC « Racines ». Ce MOOC donne des bases d'étymologie à de futurs étudiants de sciences et de santé. Il propose, sur plusieurs semaines, des cours, des exercices et un certain nombre d'outils classiques pour un MOOC (forum, FAQ, index, etc.).

Pour créer ce système d'assistance, vous allez utiliser SEPIA, le logiciel issu du [projet AGATE](#).

Le projet **AGATE** vise à étudier comment faciliter l'utilisation de logiciels par l'adjonction de systèmes d'assistance épiphytes (c'est-à-dire des systèmes d'assistance qui se greffent sur un logiciel existant, nommé application-cible, sans en perturber le fonctionnement). Le système **SEPIA** opérationnalise cette idée. Il se compose d'un éditeur d'assistance et d'un moteur d'assistance. L'**éditeur d'assistance** est l'outil dédié aux concepteurs d'assistance pour créer des systèmes d'assistance. Les systèmes d'assistance créés sont ensuite exécutés par le **moteur d'assistance** pour fournir de l'assistance lors de l'exécution de l'application-cible pour les utilisateurs finaux. Commencez par regarder la vidéo de démonstration de SEPIA *Utilisation_Generale.mp4*.

La communication entre l'éditeur et le moteur d'assistance est rendue possible par un langage pivot, le **langage aLDEAS** (cf. figure ci-dessous). Ainsi, un système d'assistance est défini par un ensemble de règles aLDEAS et leur enchaînement.

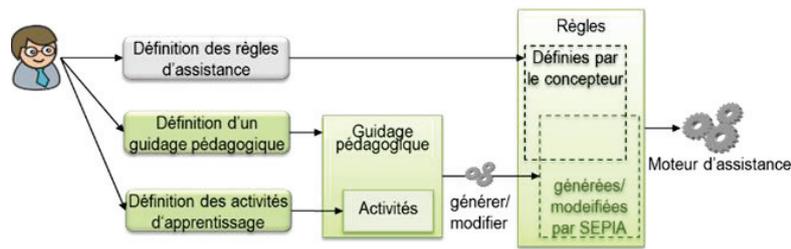


Une **règle** commence par un **événement déclencheur** (une action de l'utilisateur, un événement lié à l'assistance, une durée). Dès que cet événement a lieu, les **actions d'assistance** qui suivent (blocs de règles, règles, actions élémentaires telles que message, mise en valeur, ajout d'un bouton d'aide...) sont lancées soit immédiatement, soit en fonction d'une **condition**. Une condition prend la forme d'une **consultation** (de l'état de l'application, de l'historique de l'assistance...) avec des alternatives associées chacune à une action. Enfin, la règle peut se terminer par un **événement de fin** qui met fin à

toutes les actions élémentaires déclenchées par cette règle.

Le guidage pédagogique dans SEPIA

Dans un contexte éducatif, un tel système d'assistance peut servir à proposer un **guidage pédagogique** à l'apprenant. Les activités proposées peuvent être organisées aux apprenants de façon séquentielle, personnalisable, temporelle ou contextuelle.



Dans SEPIA, les concepteurs peuvent créer les objets pédagogiques (activité, guidage pédagogique). C'est le système qui génère et modifie ensuite automatiquement les éléments aLDEAS correspondants.

Pour mettre en place un guidage pédagogique, il faut définir dans SEPIA les activités concernées, puis le guidage pédagogique lui-même.

1. Définition des activités (cf. Figure 177) :

- Informations générales : nom de l'activité, description
- États de sortie : une activité se termine avec un ou plusieurs états de sortie possibles (cf. 2 et 3 dans la Figure 177). Il faut déclarer cet état de sortie et l'échelle des valeurs correspondante (cf. 4 dans la Figure 177) : textuelle (par exemple, une activité est terminée avec la valeur « succès » ou « échec »), numérique (par exemple, une activité est terminée avec une note sur 20).
- Règles associées : elles permettent l'assistance pendant l'activité et la clôture de cette activité avec un ou plusieurs états de sortie possibles (cf. 5 dans la Figure 177). Une règle est associée soit à aucune activité soit à une activité (cf. 6 dans la Figure 177). Pour qu'une règle termine une activité, il faut utiliser l'action déjà existante « arrêter une activité avec des états de sortie » (cf. 7 Figure 177). On devra donc préciser la valeur des états de sortie pour cette action (cf. 8 dans la Figure 177).
- **Définition du guidage pédagogique** (cf. Figure 176) :
 - Informations générales : nom, description
 - Scénarisation du guidage : placer les activités et faire le lien entre elles dans le guidage (depuis le tableau des activités)
 - Contraintes sur les états de sortie de l'activité précédente (cf. 1 dans la Figure 176))

Quand deux activités A1 et A2 conduisent à une même activité A3, la contrainte pour proposer A3 est une combinaison « et » des contraintes concernant les états de sortie de A1 et A2 (c'est-à-dire que A3 est proposée si les états de sortie de A1 **et** A2 sont ceux attendus). Pour permettre une combinaison « ou » (c'est-à-dire que A3 est proposée si l'état de sortie de A1 ou celui de A2 est celui attendu), il faut instancier l'activité deux fois. Par exemple dans la Figure 176), dans le guidage (1), l'activité A4 est proposée si les activités A1 et A2 sont toutes deux réalisées avec échec. Dans le guidage (2), l'activité A2 est proposée si l'activité A1 est réalisée avec succès ou si l'activité A3 est réalisée avec échec.

- Contraintes sur les conditions : consultation du temps, du profil ou les états de sorties des activités (y compris l'activité elle-même, pour éviter la proposition d'une activité déjà faite ou réussie). Par exemple, la condition C5 contraint la proposition de l'activité A1 (cf. 3 dans la Figure 176)).

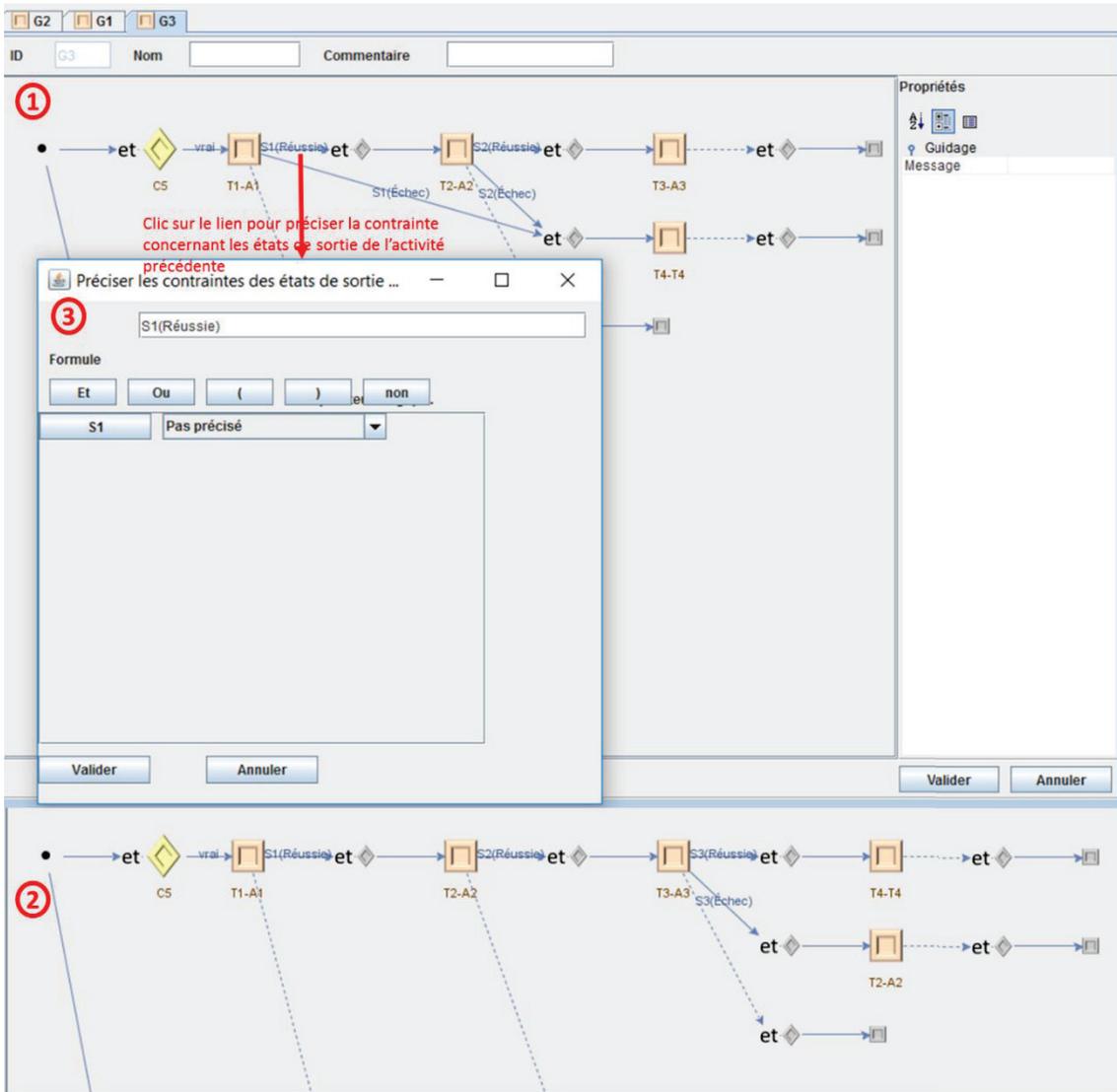


Figure 176 : création d'un guidage pédagogique

Editeur d'assistance - Création de règles d'assistance

Fichier Edition

Retour **Création d'un système d'assistance pour Fun_Mooc** Suivant

T2

1

ID: T2 Nom: A2 Commentaire:

Les règles associées à cette activité

Nom	Evènement déclencheur	Condition	Action
R11	Lancement de l'activité		A13, A18, A20
R12	E6 : mouseClicked sur le composant 115	C18	A21, A1 ou A21, A1

5

Liste des états de sortie

ID	Nom	Commentaire
S2	A2_Réussie	

2

3 Définir un état de sortie

ID: S4 Nom: S4 Commentaire: Echelle: 1-Réussie/Échec

4 Définir une échelle

ID: 2 Nom: Type: textuel Valeurs: Option 1, Option 2, Option 3, Option 4

6

8 Précision des valeurs des états de sortie d'une activité

Clic sur l'action « arrêter l'activité » pour préciser les valeurs des états de sortie de cette activité

7

Événements Conditions Actions Règles Blocs Activités

Blocs

Nom	états de sortie	
T1 : A1	S1,	R7, R8,
T2 : A2	S2,	R11, R12,
T3 : A3	S3,	R14,

G2 G1 G3 R12 R7 R11 R8

ID: R12 Nom: appartenant à l'activité: T2-A2

Événements Conditions Actions Règles Blocs

Actions d'assistance

Nom	Type	Breve description
A1	arrêt de l'activité	arrêter une activité avec un état de sorties
A2	lancement du g...	Lancer le guidage pédagogique pour proposer les activités pédagogiques
A13	message	Message = Cliquer sur cet onglet pour commencer l'activité
A14	mise en valeur	mettre en valeur le composant 52
A17	message	Message = L'activité A1 est finie
A18	mise en valeur	entourer le composant 53
A19	mise en valeur	entourer le composant 54
A20	mise en valeur	entourer le composant 115

Figure 177 : Création d'une activité

Étape 1 : compréhension d'un système d'assistance existant dans un MOOC

Dans l'éditeur d'assistance de SEPIA, sélectionnez le logiciel « MOOC Fun », puis chargez le fichier « guidage_pedagogique.xml » : il s'agit de la modélisation d'un système d'assistance pour le MOOC « Racines ».

Dans SEPIA, étudiez ce système d'assistance en **examinant** le **guidage pédagogique**, les **activités** et les **règles** dans l'éditeur d'assistance **sans exécuter** ce système dans le moteur d'assistance. Décrivez ci-dessous le guidage pédagogique que fournit ce système d'assistance :

Remarques : pour simplifier, l'activité concernant l'entraînement est réussite si la deuxième question est réussite « L'ostéoblaste est la cellule précurseur des ostéocytes qui sont des cellules osseuses. Comment appelleriez-vous la cellule précurseur des cellules sanguines que sont les hématocytes (globules rouges) ? Répondez avec un seul mot. ». L'activité concernant le bilan du jour est réussite si la première question est réussite « Littéralement, le trophoblaste nourrit l' ». Les autres activités sont réussites si l'apprenant confirme qu'il a regardé la vidéo, la fiche du cours.

? Avez-vous réussi facilement à comprendre le système d'assistance ? oui non

• Notez vos difficultés éventuelles : _____

Étape 2 : modification d'un système d'assistance existant dans un MOOC

Avant de continuer, demandez à l'encadrant la description du système d'assistance que vous venez d'examiner.

Le système d'assistance guide vers 3 activités l'une après l'autre (A1>A2>A3) :

- activité A1 : l'apprenant doit visionner la vidéo du jour 3 « Racines Blast(o)- et Germ- »
- activité A2 : l'apprenant doit faire des exercices dans la partie d'entraînement du jour 3
- activité A3 : l'apprenant doit visionner une vidéo du jour 3 « Racine Erg(o)-, Org-, Urg- ».

Demandes pour la première modification

Dans un autre contexte, l'enseignant veut enrichir ce guidage pédagogique en ajoutant 3 nouvelles activités :

- activité A4 : l'apprenant doit étudier la fiche « expresse Kin(é)- et Cin(é)- » du jour 3
- activité A5 : l'apprenant doit faire le bilan du jour 3

Modifiez le système d'assistance pour ajouter ces 2. Validez les modifications avant de tester la nouvelle version de l'assistance. Pour tester avec le moteur Web SEPIA, il faut utiliser le fichier d'assistance guidage_pedagogique_generation.xml au lieu de guidage_pedagogique.xml.

? Avez-vous réussi à effectuer cette modification dans le système d'assistance ? oui non

- Si oui, combien d'essais avez-vous fait avant de réussir ?

Demandes pour la deuxième modification

En cas de l'échec à l'activité A2, le système d'assistance doit diriger l'apprenant vers l'activité A1.

Modifiez le système d'assistance pour gérer les changements en cas d'échec. Validez les modifications avant de tester la nouvelle version de l'assistance. Pour la tester avec le moteur Web SEPIA, il faut utiliser le fichier d'assistance guidage_pedagogique_ **generation**.xml (généralisé automatiquement par SEPIA) au lieu de guidage_pedagogique.xml.

? Avez-vous réussi à effectuer cette modification dans le système d'assistance ? oui non

- Si oui, combien d'essais avez-vous fait avant de réussir ?

Étape 3 : proposition d'un guidage pédagogique pour votre projet

Imaginez maintenant un **guidage pédagogique** pour le logiciel éducatif de votre **projet**. Décrivez ce guidage de façon textuelle ci-dessous.

? Pensez-vous pouvoir définir ce guidage pédagogique avec le patron de guidage pédagogique de SEPIA ?

oui (Dessinez votre guidage pédagogique) non (Expliquez pourquoi)

Étape 4 : questionnaire

Estimez la facilité de compréhension d'un système d'assistance.

très difficile difficile neutre facile très facile

Décrivez les raisons :

Estimez la facilité de définition d'un système d'assistance.

très difficile difficile neutre facile très facile

Décrivez les raisons :
