People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 Mai 1945 Guelma

Faculty of Mathematics, Computer Science, and Sciences of Matter
Department of Computer Science
Laboratory of Science and Technology of Information and Communication

DISSERTATION

In Partial Fulfillment of the Requirements for The Doctorate degree in 3rd cycle

Domain: Mathematics and Computer Science. **Field**: Computer Science
Specialty: Computer Science

Presented by:

Rochdi BOUDJEHEM

Entitled

An agent-based approach to predict the behaviors of learners with difficulties inside human learning environments

Defended: 03/11/2022. Before, the examination board composed of:

Pr. Hamid Seridi	**Professor**	Univ. of 8 Mai 1945, Guelma	Chairman
Pr. Yacine Lafifi	**Professor**	Univ. of 8 Mai 1945, Guelma	Supervisor
Pr. Tahar Bouhadada	**Professor**	Univ. of Badji Mokhtar, Annaba	Examiner
Dr. Brahim Farou	**Senior Lecturer**	Univ. of 8 Mai 1945, Guelma	Examiner
Dr. Khaled Halimi	**Senior Lecturer**	Univ. of 8 Mai 1945, Guelma	Examiner
Dr. Zohra Mehenaoui	**Senior Lecturer**	Univ. of 8 Mai 1945, Guelma	Invited Member

Academic Year: 2021-2022

I dedicate this dissertation work to my family and friends…

*Special gratitude to my loving **Parents**, whose words of encouragement and push for tenacity still ring in my ears; I will always appreciate all they have done.*

*I also dedicate this dissertation to my Beloved **Wife** and our three children: **Aridj**, **Aous** and **Zeyd**.*

*I do not forget my dearest brothers **Badreddine**, **Hani**, and **Djalil**, their wives, and children.*

*I also dedicate this work to all my friends, especially **Mahmoud Bouziza**, **Abderrezzaq Aggoune**, **Ala Eddine Benrazek**, and **Bilel Redjimi** who have never left my side and are very special.*

Acknowledgements

This long adventure finally ends with this dissertation. It would not have been possible to finish it without God's help first, then the support of many people. Contrary to some beliefs, research is not something you do alone but in cooperation with others. One needs a lot of support and guidance from colleagues, family, and friends.

I start by thanking almighty Allah for all the blessings he bestows upon us and for his help and guidance.

Mother and Father, without you, I would not be standing here today, you supported me back when I was young, and you were always there for me when I needed you the most.

I wish to thank my supervisor, Pr. Yacine Lafifi for encouraging me to start this adventure and igniting my interest in the fields of e-learning and learning difficulties. I would not be the researcher I am today without his help and guidance. From the beginning, he has challenged me to work harder, aspires higher, and do better research. I am fortunate to have had such a great supervisor who used to be a good colleague.

I would sincerely like to thank all the examination board members, Pr. Hamid Seridi, Pr. Tahar Bouhadada, Dr. Brahim Farou, Dr. Khaled Halimi, and Dr. Zohra Mehenaoui for lending me some of their time and effort to provide me with constructive feedback on this dissertation.

I also feel honored to have my friend and brother Mahmoud Bouziza on my back. I thank him for all the help and support during the totality of the time of this project. My thanks also go to my friends Abderrezzaq Aggoune and Bilel Redjimi.

A special thank goes to Mr. Ali Benbakha my supervisor at the "ODEJ" facility, for all the support and encouragement he provided me during the whole time of this project. I also thank Mr. Mohamed Ellaggoune for all the administrative facilities he provided me. I also thank all my colleagues in the "ODEJ" facility.

A big thank you to all the members of the LabSTIC Laboratory, on their head Pr. Hamid Seridi and Ms. Madiha Kharoubi, who was very helpful. I do not forget my friend Dr. Ala Eddine Benrazek, with whom I shared most of my time in the lab and who helped me a lot.

My thanks also go to my colleagues at the Computer Science Department at Guelma University for all the support and encouraging words over the years. I would also like to thank the 2nd-year computer science students (2019/2020) and (2020/2021) for their help in testing parts of the project.

Finally, I would like to thank my brother Pr. Djalil Boudjehem, for always being there for me when I needed him.

Thank you, My Wife, for being there while I am not. Our two young children have all come to us during this project and made everything so much more fun. Aridj, Aous, and Zeyd, I am very sorry for being an absent dad all this time.

I will conclude by thanking you with all my heart.

Abstract

During their learning process, learners may encounter learning difficulties that may affect the quality of their academic outcomes. These difficulties could be triggered by external factors such as inadequate teaching content or internal factors closely related to some learners' specific characteristics. However, rather than looking for flaws in learners, it is more effective to explore extrinsic factors such as the form and relevance of teaching pedagogical content that is more adaptable to change and improvement than learner-specific factors.

Struggling learners need different forms of support to learn more effectively and, if possible, catch up with their ordinary peers in terms of academic success. Nevertheless, it is essential to identify learners with difficulties at early stages to benefit from the appropriate support. However, before that, it is imperative to determine the signs and indicators to identify these learners in the face-to-face learning context in general and in e-learning environments in particular.

This research is situated in this context and focuses on the learning difficulties faced by learners when using distance learning systems, as well as the intelligent tools available to help them overcome these difficulties. The use of distributed artificial intelligence techniques and, in particular, intelligent agents can solve the problem of detecting learners' learning difficulties and offer them the appropriate support at the right time. In recent years, new intelligent tools adopting new learning theories are continually being integrated into modern learning systems through predictive modeling used as Early Warning Systems (EWS), where one can identify and predict learners at risk in a given learning unit and inform both the teacher and the learners concerned. By collecting and analyzing learners' behavior through the traces left by them and using Distributed Artificial Intelligence (DMI) algorithms such as Multi-Agent Systems (MAS), it is possible to model, track, and monitor separately the current or even future behavior of each learner and identify which of them are doing well and which will face the likely difficulties providing valuable time to intervene and help these learners. Learners.

To achieve these goals, a set of cognitive agents have been designed and implemented to detect learners' difficulties on the one hand and predict learners' failure or success on the other hand based on their behaviors. Prototypes validating the ideas proposed in this research work were developed and tested under real learning conditions. The results obtained are considered very promising and very encouraging.

Keywords Learning difficulties, At-risk learners, Early Warning System, Learning Difficulties prediction

Résumé

Durant leur processus d'apprentissage, les apprenants peuvent rencontrer des difficultés d'apprentissage qui peuvent affecter la qualité de leurs résultats pédagogiques. Ces difficultés pourraient être provoquées par des facteurs externes comme l'inadéquation du contenu d'enseignement ou internes qui sont en étroite relation avec certaines caractéristiques de l'apprenant lui-même. Plutôt que de chercher des lacunes chez les apprenants, il est plus efficace d'explorer des facteurs extrinsèques tels que la forme et la pertinence du contenu pédagogique d'enseignement qui sont plus adaptables au changement et à l'amélioration que les facteurs spécifiques à l'apprenant.

Les apprenants en difficulté ont besoin de différentes formes de soutien pour apprendre plus efficacement et, si possible, rattraper leurs pairs ordinaires en termes de réussite scolaire. Cependant, il est essentiel d'identifier les apprenants ayant des difficultés en stages avancés afin qu'ils puissent bénéficier du soutien approprié. Mais, avant cela, il est impératif de déterminer les signes et les indicateurs permettant d'identifier ces apprenants dans le contexte d'apprentissage présentiel en général et dans les environnements d'apprentissage en ligne en particulier.

Cette recherche se situe dans ce contexte et se concentre sur les difficultés d'apprentissage rencontrées par les apprenants lors de l'utilisation des systèmes d'apprentissage à distance ainsi que les outils intelligents disponibles pour les aider à surmonter ces difficultés. L'utilisation des techniques de l'intelligence artificielle distribuée et en particulier les agents intelligents peuvent résoudre le problème de la détection des difficultés d'apprentissage des apprenants et leur offrir le soutien approprié au temps opportun.

Ces dernières années, de nouveaux outils intelligents adoptant les nouvelles théories d'apprentissage sont continuellement intégrés dans les systèmes d'apprentissage modernes grâce à la modélisation prédictive utilisée comme des Systèmes d'Alerte Précoces (Early Warning System (EWS)) où on peut identifier et prédire les apprenants à risque dans une unité d'apprentissage donnée et informer à la fois l'enseignant et les apprenants concernés. En collectant et analysant le comportement des apprenants via les traces laissées par eux et en se servant d'algorithmes d'Intelligence Artificielle Distribuée (IAD) tels que les Systèmes Multi-Agents (MAS), il est possible de modéliser, suivre, et surveiller séparément le comportement actuel ou même futur de chaque apprenant et d'identifier lesquels d'entre eux se portent bien et lesquels seront confrontés aux difficultés probables fournissant un temps précieux pour intervenir et aider ces apprenants.

Pour atteindre ces objectifs, un ensemble d'agents cognitifs ont été conçus et implémentés afin de détecter les difficultés des apprenants d'un côté et prédire l'échec ou le succès des apprenants d'un autre côté en se basant sur leurs comportements. Des prototypes validant les idées proposées dans ce travail de recherche ont été développés et testés dans des conditions réelles d'apprentissage. Les résultats obtenus sont jugés très prometteurs et très encourageants.

Mots-clés difficultés d'apprentissage, apprenants à risque, système d'alerte précoce, Prediction des difficultés d'apprentissage

ملخص

يواجه المتعلمون بشكل مستمر صعوبات في التعلم تؤثر على جودة نتائجهم الأكاديمية. يمكن أن تنجم هذه الصعوبات عن عوامل خارجية مثل عدم كفاية محتوى التدريس أو عوامل داخلية تكون مرتبطة بخصائص المتعلمين أنفسهم. ولكن بدلاً من البحث عن نقاط الضعف لدى المتعلمين، قد يكون من الأكثر فعالية استكشاف العوامل الخارجية مثل شكل ونوعية المناهج التعليمية أو تقنيات التدريس التي تكون عادة أفضل قابلية للتكيف مع التغيير والتحسين أكثر من العوامل الخاصة بالمتعلم.

قد يحتاج المتعلمون الذين يعانون من صعوبات في التعلم إلى أشكال مختلفة من الدعم أكثر من غيرهم للتعلم بشكل أكثر فعالية، وإذا أمكن، اللحاق بأقرانهم من حيث التحصيل الدراسي. ولهذا كان من الأهمية بمكان تحديد المتعلمين الذين يعانون من صعوبات في التعلم في مراحل مبكرة حتى يتمكنوا من الاستفادة من الدعم المناسب. ولكن قبل ذلك، من الضروري تحديد العلامات والمؤشرات التي تؤدي إلى التعرف على هؤلاء المتعلمين، وخاصة في بيئات التعلم عن بعد.

يقع هذا البحث في هذا السياق ويركز على صعوبات التعلم التي يواجهها المتعلمون عند استخدام أنظمة التعلم عن بعد وكذلك الأدوات الذكية المتاحة لمساعدتهم على التغلب على هذه الصعوبات.

في الأساس، يهتم هذا البحث بتأثير استخدام تكنولوجيا الوكيل الذكي لإعادة تعديل المناهج التعليمية، وبالتالي مساعدة المؤسسات التعليمية على تحسين جودة خدماتها.

يمكن استخدام تقنيات الذكاء الاصطناعي الموزع ، وخاصة تكنولوجيا الوكيل الذكي لحل مشكلة التنبؤ بصعوبات التعلم للمتعلمين وتقديم الدعم المناسب لهم في الوقت المناسب.

و مع التطور التكنولوجي يتم باستمرار دمج العديد من الأدوات الذكية الجديدة التي ترتكز على أحدث نظريات التعلم في أنظمة التعلم الحديثة من خلال النمذجة التنبؤية المستخدمة كأنظمة إنذار مبكر(EWS)، حيث صار من الممكن تحديد وتنبؤ سلوك المتعلمين المعرضين للخطر في وحدة تعليمية معينة وإبلاغ كل من المعلم والمتعلمين، و يتم ذلك من خلال جمع وتحليل آثار المتعلمين باستخدام خوارزميات الذكاء الاصطناعي الموزع (DAI) مثل الأنظمة متعددة الوكلاء(MAS)، حيث صار بالإمكان نمذجة وتتبع ومراقبة السلوك الحالي أو المستقبلي لكل متعلم على حدى وتحديد أي منهم سيتلقى تعليمه بشكل جيد ومن منهم يحتمل أن يواجه صعوبات. من خلال تحقيق ذلك، يمكننا توفير وقت ثمين للتدخل ومساعدة هؤلاء المتعلمين.

لتحقيق هذه الأهداف، تم تصميم ووضع قيد التشغيل مجموعة من الوكلاء الأذكياء (Intelligent Agents) للكشف عن صعوبات المتعلمين من ناحية والتنبؤ بفشل أو نجاح المتعلمين من ناحية أخرى وذلك بناءً على سلوكياتهم. إضافة إلى ذلك، تم تطوير نماذج أولية للتحقق من صحة الأفكار المقترحة في هذا العمل البحثي واختبارها في ظل ظروف تعلم حقيقية. كانت النتائج التي تم الحصول عليها واعدة ومشجعة للغاية.

كلمات مفتاحية: صعوبات التعلم، المتعلمون المعرضون للخطر، نظام الإنذار المبكر، التنبؤ بصعوبات التعلم.

Contents

Acknowledgements . . . iii
Abstract . . . v
Résumé . . . vii
ملخص . . . ix

Contents xi

List of Figures xv

List of Tables xviii

Introduction 1

I State of the Art 7

1 **Intelligent Tutoring Systems and E-Learning Systems** 9
1.1 Introduction . . . 10
1.2 Intelligent Tutoring Systems . . . 10
1.2.1 Computer Assisted Instruction . . . 11
1.2.2 Intelligent Tutoring Systems (Intelligent Computer-Assisted Instruction) . . . 12
1.2.3 Basic Models of Intelligent Tutoring Systems . . . 14
1.2.4 Developped Intelligent Tutoring Systems . . . 16
1.2.5 Limitations of Intelligent Tutoring Systems . . . 17
1.3 E-Learning Systems . . . 17
1.3.1 What is E-Learning? . . . 17
1.3.2 E-Learning Terminology Confusion . . . 19
1.3.3 Traditional, Online Education and Blended Learning . . . 23
1.3.4 Why do so many people dislike E-Learning? . . . 23
1.4 Conclusion . . . 23

2 **Learning Difficulties: Diagnosis, Detection, and Prediction** 25
2.1 Introduction . . . 26
2.1.1 What are Learning Difficulties? . . . 27
2.1.2 Learners with Difficulties: "At-risk" and "Struggling" Learners . . . 27

2.1.3 *Learning Difficulties indicators* 28
2.2 Learning Difficulties Factors and Causes and Effects on the Learner 30
2.2.1 *Teaching-Related Factors (Teaching Methods and Strategies)* . . 31
2.2.2 *Curriculum-Related Factors (Domain Knowledge)* 33
2.2.3 *Learner-Related Factors* 34
2.2.4 *Classroom's Environment-Related Factors (Cognitive, Emotional, Behavioural, and Social)* 38
2.2.5 *Importance of Early Identification of Learning Difficulties* . . 39
2.3 Identifying Learners with Difficulties inside E-Learning Environments 39
2.3.1 *Detection of Learners with Difficulties using Artificial Intelligence* 40
2.3.2 *Existing Detection, Prediction, and Intervention Systems* . . 40
2.3.3 *Used Artificial Intelligence Techniques to Detect and Predict Learning Difficulties* 43
2.4 Conclusion 44

3 Multi-Agent Systems In Human Learning Environments **47**
3.1 Introduction 48
3.2 Distributed Artificial Intelligence 48
3.2.1 *What is Artificial Intelligence?* 49
3.2.2 *Distributed Artificial intelligence (DAI)* 50
3.3 What is an agent? 50
3.3.1 *Definitions* 51
3.3.2 *Intelligent/Rational Agents* 55
3.3.3 *Classification of Agents* 56
3.3.4 *Reinforcement Learning Agents* 61
3.3.5 *Pedagogical Agents* 61
3.4 Multi-Agent Systems 64
3.4.1 *Definition* 65
3.4.2 *The Control Problem* 65
3.4.3 *Mechanisms of Coordination* 66
3.4.4 *Existing Multi-Agent System Platforms* 67
3.5 Intelligent Agents and Human Learning 67
3.5.1 *Usage of Intelligent Agents in Education* 68
3.6 Conclusion 72

II PROPOSED APPROACH AND RESULTS VALIDATION **75**

4 Proposed Agent-based EWS to Detect and Predict At-Risk Learners **77**
4.1 Introduction 78
4.2 Problematic of the Research 78

4.3 Contributions of the research 79
4.4 The Proposed Course Structure 81
4.5 Contribution I: Proposition of a New MAS to Detect At-Risk Learners 82
4.5.1 Subsystem 1: The Learner Model Update Subsystem . . . 84
4.5.2 Subsystem 2: The Difficulty Detection Subsystem 86
4.5.3 Subsystem 3: The Intervention Subsystem 95
4.5.4 Intelligent Agents 95
4.6 Contribution II: Extending the proposed MAS to predict At-Risk Learners 97
4.6.1 Subsystem 1: The Learner Model Update Subsystem . . . 98
4.6.2 Subsystem 2: The Difficulty Detection Subsystem 100
4.6.3 Subsystem 3: The Difficulty Prediction Subsystem 100
4.6.4 Subsystem 4: The Intervention Subsystem 103
4.6.5 Intelligent Agents 103
4.7 Proposed Agents Details 104
4.7.1 The Learner Model Update Subsystem: 105
4.7.2 The Difficulty Detection Subsystem: 105
4.7.3 The Prediction Subsystem 107
4.7.4 The Intervention Subsystem 108
4.7.5 Scenario of Learning Difficulties Detection and Prediction Using MAS 109
4.8 Conclusion 110

5 Prototypes implementation, Experiments Results, and Discussions 111
5.1 Introduction 112
5.2 Building Prototypes 112
5.2.1 Development Tools to Build the Prototypes 112
5.3 LearnDiP+ Functionalities 115
5.3.1 LearnDiP+ for the Students 115
5.3.2 LearnDiP+ For the Teachers 115
5.4 Accessing and Using LearnDiP+ 117
5.4.1 Teacher Space 120
5.4.2 Admin Space 124
5.4.3 Student Space 125
5.5 Experimentations, Results and Discussion 130
5.5.1 Experimentation I: Testing the effectiveness of the prototype to detect at-risk learners and intervene to enhance their cognitive level and diminish dropouts number 131
5.5.2 Experimentation II: Testing the effectiveness of the prototype to predict struggling learner dropouts 134
5.6 Conclusion 137

Conclusion and Future Work 139

Author's Publications 143

Bibliography 145

A Existing Multi-Agent System Platforms 165
A.1 JADE (Java Agent DEvelopment Framework) 165
A.2 MadKIT (Multi-Agent Development KIT) 166
A.3 JACK Intelligent Agents 166
A.4 Agent Builder 166
A.5 Cougaar (Cognitive Agent Architecture) 167
A.6 Zeus Agent Building Toolkit 168
A.7 MASON 168
A.8 Other MAS Platforms 169

B Types of Agents' Environment 171
B.1 Accessible vs. inaccessible 171
B.2 Deterministic vs. non-deterministic 171
B.3 Episodic vs. non-episodic 171
B.4 Static vs. dynamic 172
B.5 Discrete vs. continuous 172

APPENDICES **165**

List of Figures

1.1 Computer Assisted Instruction Flow of Events (Shute and Psotka, 1994) 11
1.2 Basic Models of an Intelligent Tutoring System (Butz et al., 2006) . 14
1.3 E-Learning in the Educational Field, based on (Mason and Rennie, 2006, p. xvii) 20

2.1 Usage of the terms "At-risk" and "Struggling" in the literature as reported by Google Books Ngram Viewer 28
2.2 Learning Difficulties Factors and Probable Causes 31

3.1 Types of AI (Abioye et al., 2021) 50
3.2 An Agent as specified By Wooldridge and Jennings (1995) 51
3.3 An Agent as specified By (Ferber, 1997)(Translated from French) . 51
3.4 An Agent An Agent as specified By (a) (Russell and Norvig, 1995) and (b) (Russell and Norvig, 2010) 52
3.5 An Agent in its Environnment 52
3.6 Reactive Agent Diagram (Ferber, 1997) 56
3.7 Cognitive Agent Diagram (Ferber, 1997) 57
3.8 Simple-reflex Agent Diagram (Russell and Norvig, 2010) 58
3.9 Simple-reflex Agent with Internal State Diagram (Russell and Norvig, 2010) 58
3.10 Goal Directed Agent with explicit goals (Russell and Norvig, 2010) . 59
3.11 Complete Utility-Based Agent (Russell and Norvig, 2010) 60
3.12 Learning Agent (Russell and Norvig, 2010) 60
3.13 A Reinforcement Learning Agent (Sutton and Barto, 2012) 61

4.1 Proposed Approach to Model, Detect, Predict then Assist Learners . 80
4.2 Example of Learning Object (LO) Composition 81
4.3 The Course Structure 82
4.4 General Description of A New Approach to Detect Learning Difficulties 83
4.5 ML_i Model of the *Learner* L_i 85
4.6 Flow Diagram of the calculation algorithm of $LODL_{ijk}$ of the LO_k of the *Course* C_j of the *Learner* L_i, based on *Primary* and *Secondary* indicators 88
4.7 Simple Linear Regression Slope and Intercept 89
4.8 (a) Computed slope of regression using six LO points, (b) Computed slope of regression using three LO points 90

4.9 Example of SPLOBS (Improvement) 91
4.10 Example of SLACLO (Regression) 92
4.11 Number of Delay Days of *Learner* L_i accessing a Learning Object LO_{jk} past Deadline 93
4.12 Case 1: The *Learner* L_i did access the LO_k of the *Course* C_j . . . 94
4.13 Case 2: The *Learner* L_i did not access the LO_k of the *Course* C_j after the deadline 94
4.14 Example of SNAPLO. 95
4.15 Proposed Multi-Agent System for Learning Difficulties Detection in a given Course 96
4.16 General Description of the Approach to Detect and Predict Learning Difficulties 98
4.17 New Model of The Learner Taking into Consideration the Prediction 99
4.18 *Learner* L_i CDV after passing LO_4 99
4.19 CDV of a dropper 100
4.20 CDV of a succeeder 100
4.21 Sample of stored CDVs (Course with 10 LOs) 101
4.22 Using Cosine Similarity to find *FVs* Similar Vectors to *SV* . . . 101
4.23 Proposed Multi-Agent System for Learning Difficulties Detection and Prediction in a given Course 104
4.24 Scenario of interactions between the proposed Intelligent Agents . 109

5.1 Student Progress and difficulty Levels of a given Course . . . 115
5.2 Student Progress and difficulty Levels of a given Learning Object . 116
5.3 Student Progress and difficulty Levels of a given Learning Object . 116
5.4 Student Predictions of a given Course 117
5.5 LearnDiP+ Home Page 119
5.6 Teacher Space Welcome Page 119
5.7 Admin Space Page 120
5.8 Teacher Editable Learning Content 120
5.9 Teacher Courses Edition 121
5.10 Teacher Learning Objects Edition 121
5.11 Teacher Quizzes Edition 121
5.12 Teacher Questions Edition 121
5.13 Detected Learning Difficulties in the course listed for each student . 122
5.14 Predicted Learning Difficulties in the course listed for each student . 122
5.15 The Teacher enrolling a Student 122
5.16 Students' Progress and Difficulties in the Course and Chapters . . 123
5.17 Students' Progress and Difficulties in a Selected Chapter . . . 124
5.18 Predicted Learning Difficulties in the course listed for each student . 124
5.19 Online Students and Teachers 124
5.20 Chat between Teachers and Students 124
5.21 Adming Users Management 125

5.22 Adming Roles Management 125
5.23 Student has Access to the Enrolled Course 125
5.24 Student Course List 126
5.25 Course Content 126
5.26 Chapters Content 127
5.27 Lessons Content 127
5.28 Evaluation Access through the Chapter 128
5.29 Evaluation Access through the Lesson 128
5.30 Evaluation Attempt 128
5.31 Evaluation Scores and Attempt Number Check 129
5.32 Evaluation Access through the Lesson 129
5.33 Evaluation Access through the Chapter 129
5.34 Course Progress and Difficulties 130
5.35 Chapter Progress, Difficulty Level and Scores 130
5.36 Distribution of PreTest and PostTest values for TG and CG . . . 133
5.37 Differences in means of PreTest and PostTest values for TG and CG . 134
5.38 Independent Samples of the PreTest and PostTest values for TG and CG 134

List of Tables

1.1 Comparison Between Typical ITSs and E-Learning Systems Features 24

2.1 Dropout-related works grouped by "Purpose of Research" 42
2.2 Comparative table of the best dropout predictors 43
2.3 Most used ML Techniques to predict learners' dropouts 45
2.4 Other ML Techniques used to predict learners dropouts 46

3.1 Most popular Multi-Agent Plateforms 67

4.1 Example of the SPLOBS of a given learner 91
4.2 Intelligent Agents Distribution Among Proposed Subsystems. . . . 104
4.3 Personal Agent Features 105
4.4 Continuous Control Agent Features 106
4.5 Difficulty Detection Agent Features 107
4.6 Difficulties Prediction Agent Features 108
4.7 Intervention Agent Features 108

5.1 List of students enrolled in a given Course with all the real-time detection functionalities 117
5.2 List of students enrolled in a given Course with all the system prediction functionalities 118
5.3 Conducted experimentations. 130
5.4 Dropout statistics among TG and CG 132
5.5 Tests of Normality 132
5.6 Independent-Samples Mann-Whitney U Test Summary PreTest and PostTest across Group 133
5.7 Confusion Matrix 136
5.8 Prediction's Confusion matrix 136

B.1 Examples of environments and their characteristics. 172

Introduction

Research Background

All children attend school for the first time full of hope, motivation, and self-confidence. However, at the end of the first year, some lost confidence due to a lack of success. Failure impairs a child's sense of self-efficacy, self-esteem, and motivation (Rosner, 1994; Neal and Kelly, 2002; Slavin et al., 1994).

Educational techniques have remained consistent throughout human history, with instructors delivering knowledge to learners face-to-face. However, just as technological developments have revolutionized every element of life, from the individual to the social, they have also revolutionized contemporary educational methods. By the 1970s, the microprocessor transformed the computer industry from cabinet-sized computers to small Personal Computers (PCs).

PCs established the foundation for a tremendous information revolution, which grew in prominence with the advent of the Internet and its mainstreaming like the World Wide Web (WWW). As PCs evolved and networks spread globally, the use of Information and Communication Technologies (ICTs) grew to infiltrate all sectors of society, industry, and government. The digital revolution has brought unprecedented changes in how people access, consume, discuss, and exchange information. It was just a matter of time before ICT was implemented in educational institutions to enhance current teaching and learning processes.

Motivation for the Research

In traditional classrooms, the teacher should know if one or more learners are not receiving their instruction well or facing difficulties. Sometimes, it takes only a few minutes of conversation to assist and help these learners overcome these difficulties. That can be true for small-size classes, but how about large or massive classes like universities? Add to that some other factors like using e-learning platforms as a secondary means of education or in situations like the last Coronovirus pandemic when that kind of teaching has become mandatory. In both cases, the direct physical interaction between teachers and learners becomes hindered and less frequent. Decreased physical interaction between the learner and his peers and teachers can have severe consequences, especially when the former faces difficulties. In this case, no one is around to notice his condition and step up to help him.

Fortunately, technology could solve some of these problems the same way it causes trouble by carefully designing intelligent e-learning environments and providing them with tools enabling them to monitor and track learners closely and individually. Not only that, these environments should consider learners' emotional and affective aspects by taking advantage of the advances and breakthroughs in the fields of cognitive psychology and educational sciences to understand the learner as a human being.

In that direction, the current research is headed to find the learning difficulties in e-learning systems. On a theoretical level and the field of application retained, we boarded several fields of research discipline intervening jointly in this research with more or less strength. These disciplines are :

- Computer Science and Artificial Intelligence.
- Cognitive Psychology and Educational Sciences.

Aim of the Research

Thankfully, new learning theories emerge daily, focusing on the learner as the central actor and transforming the learning process into a more flexible, personalized, and tailored operation. They are more interested in getting feedback from learners, not just in the form of a final evaluation when it is too late to do anything in case of potential difficulties. Instead, processing the learner's early feedback can provide insight into their current or future state and even predict potential issues, allowing for intervention and thus fighting these issues and problems.

Additionally, new intelligent tools embracing these theories are continuously integrated into modern learning systems, allowing them to detect and predict such scenarios and even intervene autonomously to prevent them from happening. By acquiring and analyzing early feedback from learners, these tools can help instructors better understand their learners' condition and foresee their probable difficulties and challenges, giving the formers more time to intervene and preventing the latter from falling. Moreover, these tools allow learners to be aware of their current needs and abilities and warn them about potential future risks, thus helping them perform better by affronting these difficulties. Thanks to predictive modeling and learning analytics techniques, identifying at-risk students and predicting their learning performance in a class is now possible. A predictive model can be used as an Early Warning System (EWS) to identify and predict at-risk students in a course and inform both the teacher and the students (Sandoval et al., 2018; Howard et al., 2018; Waddington et al., 2016).

Fortunately, learners leave a large set of data about their tracks and actions during their learning process. Collecting and analyzing this valuable data using Artificial Intelligence (AI) algorithms makes it possible to track, monitor, model, and profile each learner's current or future condition separately and tell which one of them is doing well and which ones are facing problems.

As a result, teachers and instructors should be able to select from a range of intervention and communication tactics the best ones to assist and help these students improve their performance and overcome their challenges. Furthermore, these systems can be equipped with autonomous AI tools that allow them to intervene with these students without any human interaction.

problematic and addressed research questions

Setting up such systems is complex and requires the preparation of some initial conditions, like choosing the best learner's indicators and predictors that can reflect his behavior or choosing the best algorithms to identify or predict any probable difficulties. It is in this research context that our dissertation's research belongs. It aims at answering the following research questions.

- How can the most pertinent signs and indicators are found based on the learners' traces that most accurately reflect their behavior in online learning environments?
- How can these indicators be used to detect and predict potential learning difficulties early on?
- How can one plan sufficiently early and well-designed interventions based on predictions to support these learners?
- By proactively assisting these students, may one greatly improve their cognitive ability and lower dropouts numbers?

Research Methodology

To answer those questions, we first conducted bibliographic research to review the literature about our main research field and understand the conditions and emotional state of the learners who faced difficulties in traditional and electronic learning environments. We also checked the theoretical background of existing solutions proposed to fight that phenomenon. Our research also included Distributed Artificial Intelligence (DAI), especially intelligent agents, as potential tools to implement our future proposed approach.

Intelligent agents are proactive autonomous entities carrying out actions without human intervention. Many researchers in education-based teaching used these tools for different purposes. We also included these agents in that research and how they are used to serve the educational field.

Performing this theoretical research allowed us to propose a novel Agent-based Early Warning System to predict at-risk learners based on their behavior and thus intervene to save them before it is late. To validate the proposed approach, we implemented a new system called "LearnDiP+" (**Lear**ning **Di**fficulties **P**revention plus) that embraces this approach. The approach and the prototype were performed in two major steps, explained further in this manuscript.

LearnDiP+ comprises autonomous subsystems that are constantly communicating and holding specific tasks. Each subsystem delegates several cognitive agents to carry out its tasks. Even though each of these agents performs different tasks and has different objectives, they all cooperate to achieve the system's global goals.

We tested the implemented prototypes by conducting separate experiments at our university with real students to validate the proposed theoretical approach.

This research focuses on learning difficulties that learners may face during their journey to acquire knowledge. It dissects the learners' conditions and circumstances when reaching that goal while emphasizing their challenges and focusing on solutions proposed in the literature. It also examines specific challenges experienced by learners when using distance learning systems and available tools and applications to assist them in overcoming these challenges.

It also sheds light on the history and effect of ICT use in educational environments, tracing the transition from conventional face-to-face teaching to e-learning and online teaching. Additionally, it presents how these technologies improved the quality of teaching in existing educational institutions. In the end, this research discusses the technological influence of intelligent agent technology in allowing a readjustment of learning curricula and strategies and teaching methods and tactics, thus assisting educational institutions in coping with increasingly diverse learners populations.

Manuscript Overview

This manuscript is organized as follows. It begins with a General Introduction followed by two main parts. The first part is the State of the Art, while the second comprises the theoretical and practical proposals alongside experimentations results. Each part contains numbered chapters as follows:

- **Part I:** ***State of the Art:***

  This first part presents the "State Of The Art" in different research fields related to our research. It begins by providing some background on existing forms of learning systems that rely on the use of Information and Communication Technologies (ICTs), such as Intelligent Tutoring Systems (ITSs) and e-learning systems which constitute the two most known forms of that kind of learning treated in the context of this research.

  Then, it examines the problem of learning difficulties faced by learners in both traditional and technology-based learning environments, how artificial intelligence is used to diagnose and identify at-risk and struggling learners, and the existing solutions to help these learners.

  In the end, this part provides a thorough background on Distributed Artificial Intelligence (DAI) and especially intelligent agents, their structure, types, and classifications, how they can work together within Multi-Agent Systems, and most importantly, how they were used in the educational field

  This part is divided into three chapters:

  - **Chapter One (1)** explores technology-based learning environments, such as e-learning environments and intelligent tutoring systems, their history, structure, and advantages and inconveniences. It focuses on the fundamental theories of both technologies and how they have been used over the years. This chapter is concluded with a comparison between them and shows how they can be combined to gain the best of them.
  - **Chapter Two (2)** investigates the existing learning difficulties definitions according to several parties such as research scientists and official academic organizations. It highlights the signs and indicators common among learners that face these difficulties and also emphasizes their effects on the learners in different aspects. It also explains how these difficulties can be identified and how to fight them. This chapter also explores existing

artificial intelligence-based solutions to perform the previously mentioned tasks within learning environments.

- **Chapter Three (3)** introduces Artificial Intelligence (AI) and Distributed Artificial Intelligence (DAI). It focuses on intelligent agents as a form of DAI; what are they, where do they exist, what are their characteristics, how the literature classifies them, and how do they work together inside Multi-Agent Systems (MAS)? The second part of this chapter focuses on using intelligent agents and Multi-Agent Systems in online learning systems and how these tools could be the tool of choice to achieve pedagogical tasks for teachers and learners. It also presents some existing online learning systems that use that technology to help teachers and instructors carry out their pedagogical tasks.

These three chapters are addressed simultaneously from the psychological and computer science perspectives through the contributions, limitations, and issues of existing systems and works.

- **Part II:** ***Proposed Approach and Results Validation:***

  In this second part, we look more closely at the proposals we are making. We chose to take the challenge and try to design, implement and test a novel system capable of achieving all the previously mentioned goals. The following chapters provide all the necessary details about each project phase. This part is divided into two chapters:

  - **Chapter Four (4)** introduces the proposed theoretical approach to address the problems discussed in this research. The proposed approach is an Early Warning System (EWS) to detect and predict learners with learning difficulties and identify those experiencing or who will experience difficulties. Two contributions were introduced in this chapter:
    - ▷ **The first contribution** is the proposition of an agent-based approach capable of identifying learners with learning difficulties and autonomously intervening to help these learners. It focused on learning difficulties indicators and learner modeling to identify struggling learners and autonomously intervene to help them. The proposed approach computes a "difficulty level" for each course and each learning object for each learner (enrolled in that course) based on a set of primary and secondary indicators used to deduce his situation regarding that course. This contribution comprises a set of abstract subsystems that delegate cognitive agents to perform the aimed tasks as a whole Multi-Agent System.
    - ▷ **The second contribution** is the proposition of an agent-based approach capable of predicting learners with difficulties at early stages so it can intervene autonomously to save those learners before it is late. This contribution extends the first contribution by adding a fourth subsystem tasked with the prediction operations. To this end, the proposed approach used computed difficulty levels gathered as a vector for each course, and each learner enrolled in that course. This vector represents the history of difficulties of this learner in this course. To achieve that, the proposed approach calculates the distance between the current learner's difficulty vector in a given course and those of his predecessor peers to look for similarities. The results are

then used to predict the situation of this learner. The new proposed subsystem delegates another set of cognitive agents to carry on its tasks.

This chapter explains in detail the structure, role, objectives, and functions of each intelligent agent in the two proposed contributions. It illustrates with a scenario the different interactions between the proposed MAS and the different human actors to perform the detection, prediction, and intervention tasks.

- **Chapter Five (5)** presents the two prototypes developed, "LearnDiP" and "LearnDiP+," used to test and validate the theoretical approaches presented in the previous chapter. It introduces the different technologies used to develop these prototypes. Furthermore, it presents the main functionalities they provide to all the actors in the learning process: learners, teachers, and administrators. It presents the spaces of each of these actors, exposes the available functions, and illustrates them with some screenshots. This chapter also presents the experiments conducted with these two prototypes. It highlights the results obtained through the experiments to validate the proposed approaches to detect, predict and prevent learning difficulties in human learning environments.

This manuscript is concluded with withdrawn conclusions and future work, followed by the References.

Part I

State of the Art

Chapter 1

Intelligent Tutoring Systems and E-Learning Systems

Contents

1.1 Introduction 10
1.2 Intelligent Tutoring Systems 10
1.2.1 *Computer Assisted Instruction* 11
1.2.2 *Intelligent Tutoring Systems (Intelligent Computer-Assisted Instruction)* 12
1.2.3 *Basic Models of Intelligent Tutoring Systems* 14
1.2.4 *Developped Intelligent Tutoring Systems* 16
1.2.5 *Limitations of Intelligent Tutoring Systems* 17
1.3 E-Learning Systems 17
1.3.1 *What is E-Learning?* 17
1.3.2 *E-Learning Terminology Confusion* 19
1.3.3 *Traditional, Online Education and Blended Learning* . . . 23
1.3.4 *Why do so many people dislike E-Learning?* 23
1.4 Conclusion 23

1.1 Introduction

Over the years, the development of two major educational technologies that use digital instructional tools to support student learning has progressed widely and independently; intelligent tutoring systems (ITS) and e-learning. Both ITS and e-learning have advantages and disadvantages.

ITSs are domain specialized and rely on precise knowledge and learner modeling. In contrast, e-learning systems can be used in various situations and focus on connecting the learner to both the knowledge content and his peers. They have primarily focused on promoting and facilitating problem-solving in learning. Typically, they are based on customized, rich information representations and respond to the needs of the learners through cognitive diagnosis and user modeling techniques (Brooks et al., 2006).

On the other hand, the word "eLearning" encapsulates several fundamental concepts: learning activities facilitated by Web-based technologies such as learning management systems (LMS) like WebCT and Moodle, conferencing and discussion systems, and rich multimedia materials. E-Learning systems cover a wide range of qualities. However, they have mostly been criticized for lacking pedagogical and psychological relevance and the lack of controlled evaluations.

We think the integration and partnership between the two communities should undoubtedly benefit both. Despite this, there has not been much exchange of ideas and technologies between the two groups because of their different cultures.

This chapter presents both technologies, their history, aims, and objectives, and also how the two of them have been used over the years. It also provides the fundamental theories these technologies are build-on. Furthermore, it presents common grounds between them and how one can benefit from the advantages of the other to cover its shortcomings.

1.2 Intelligent Tutoring Systems

Pressey (1926), an educational psychology professor at Ohio State University, created a machine in the early 1920s to deliver drill and practice items to students in his introductory classes. Each successful answer earned the learner candy from the machine. Pressey's system was sophisticated enough at the time to show preselected questions and answers. Even though it was a rigid machine that only rewarded candy for correct answers, it included some current educational techniques (Fry, 1960).

General-purpose computers appeared around 1950, establishing artificially intelligent machines. Turing (1950), a mathematician, logician, and computer scientist, linked computers and intelligence by inventing the Turing test, utilized to discriminate between humans and machines based on questions asked of both, and that the computer must communicate like a human to be effective (Shute and Psotka, 1994). In the mid-1950s, Educational psychologists began customizing tutoring tools to give a practical learning experience for humans (Fry, 1960). Intelligent Tutoring Systems (ITSs) have been suggested as a potential method of delivering these adaptive tutoring solutions since the 1970s (Shute and Psotka, 1994).

This section presents how Intelligent Tutoring Systems appeared and progressed over time from simple "Programmed Instructions" to fully working ITSs.

1.2.1 Computer Assisted Instruction

In the mid-1960s, Programmed Instruction (PI) became popular. The designer must decide on the input and output for PI. Before moving on to the next question, students who respond erroneously will be corrected immediately and informed of their errors. When computers began to be used extensively in Education, many acronyms were used to reflect different uses, such as CAI, CAL, CBE, CBL, CMI, and others (see Example 1) (O'Shea, 1979).

Example 1.

$$C = Computer \begin{cases} \textbf{CAI} \text{ Computer Assisted Instruction,} & \text{A=Assisted | Aided} \\ \textbf{CBE} \text{ Computer Based Education,} & \text{B=Based | M=Managed} \\ \textbf{CMI} \text{ Computer Managed Instruction,} & \text{I=Instruction} \\ \textbf{CAL} \text{ Computer Assisted Learning,} & \text{L=Learning} \\ \textbf{CBL} \text{ Computer Based Learning,} & \text{E=Education} \end{cases}$$

Shute and Psotka (1994) defined CAI as a method of teaching that incorporates PI but is handled by computers where instructors build all program activities and possible paths in advance.

As shown in Figure 1.2, the standard CAI program begins by presenting some content to the learner, followed by representing a problem that constitutes a portion of the curriculum to be solved. The CAI program evaluates whether the student's answer is right or wrong at every point in the curriculum and moves the student to the proper path by comparing their answers to the correct ones and providing relevant feedback afterward (Shute and Psotka, 1994).

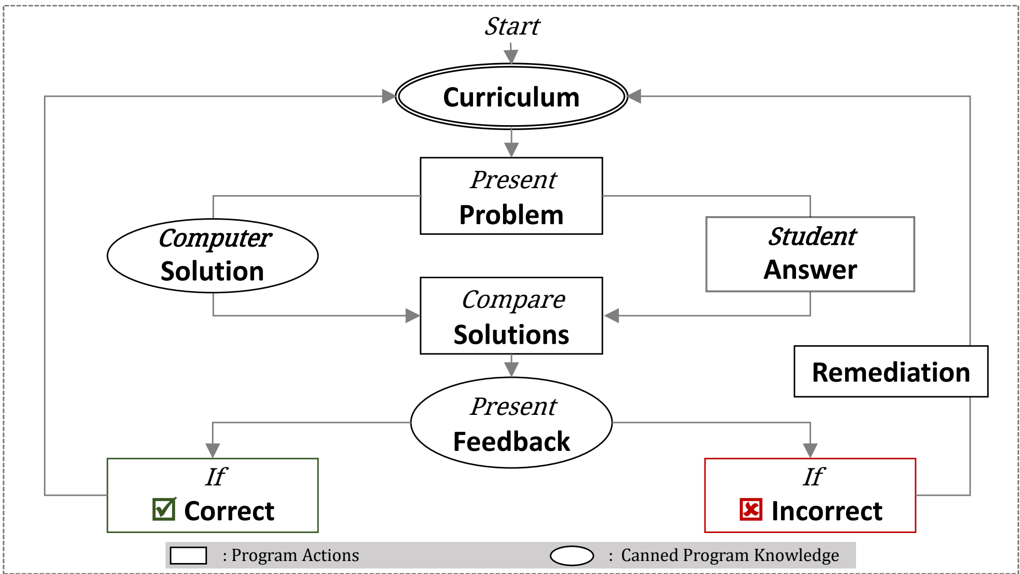


Figure 1.1 – *Computer Assisted Instruction Flow of Events (Shute and Psotka, 1994)*

If learners answer correctly, they are moved ahead in the curriculum, and a new problem is picked and provided. However, if the answer is erroneous, remediation is triggered, which examines earlier information and provides more straightforward questions that progress to the level of difficulty of the original material. Built-in remediation loops help students that attempt to answer a question incorrectly. Typically, remediation includes locating the cause of the mistake and treating it uniquely. The problem-solving module measures the student's mastery of the knowledge or ability being taught at that moment (Shute and Psotka, 1994).

According to Chambers and Sprecher (1983), CAI software applications are divided into categories such as Drill and Practice, Tutorial, Problem Solving, Simulation, Tool Software, and Computer Programming.

Unfortunately, unlike human teachers, Computer Assisted Instruction (CAI) programs do not profit from their gained expertise in the classroom. After teaching one hundred students, a CAI program that teaches poorly in some aspects may continue to teach poorly in the same way. As a result, several ideas and techniques from Artificial Intelligence were developed and applied to construct self-improving intelligent CAI programs (O'Shea, 1979).

1.2.2 Intelligent Tutoring Systems (Intelligent Computer-Assisted Instruction)

In the '70s, Artificial Intelligence (AI) was in full bloom, looking for computer and cognitive research applications. CAI was a mature and promising technology with a market. The educational system looked for ways to deal with large groups in schools. Inventors and scientists hoped that combining AI and CAI could improve educational instruction. Burns and Capps (1988) demonstrated that individual tutoring outperforms group tutoring. Meanwhile, since CAIs instruct identical information to all learners regardless of their learning capacities, CAI systems have proven ineffective.

The development of the "computer tutor" has necessitated extremely complicated computer programs, leading CAI researchers to employ artificial intelligence approaches. Artificial intelligence (AI) works well in natural language understanding, the representation of knowledge, and inference methods. Specific AI applications, such as algebraic simplification, calculus, and theorem proving, have been utilized by several researchers to create more intelligent and effective CAI programs. New AI-based CAIs were then called Intelligent Computer Assisted Instruction (ICAI). Early research on ICAI systems centered on topic matter representation. In these problem-solving conversations, a high level of domain expertise was displayed Clancey et al. (1979b).

An intelligent tutoring system (ITS)[1] suited to each student's needs, and the learning pace was seen as a good alternative by AI researchers.

Abu Naser (2016) defines an Intelligent Tutoring System (ITS) as software that intends to give learners instant and adaptive education feedback without the need for instructor intervention. The purpose of an ITS is to use various information

[1] As stated by (Fischetti and Gisolfi, 1990), the abbreviation "ICAI" emerged from CAI (Intelligent CAI) to describe educational research involving AI. However, "ICAI" has been replaced by the acronym "ITS", coined by Sleeman and Brown (1982). In this dissertation, we use only the ITS acronym to refer to both ITS and ICAI.

technologies to assist efficient learning. As reported by Fenza and Orciuoli (2016), an ITS should provide learning activities tailored to the student's knowledge and background to stimulate the learning process, provide personalized feedback, and prevent the learner from becoming frustrated or disengaged as a result of poor performance.

ITSs are computerized learning environments that strive to customize teaching and are evolved from Computer Assisted Instruction (CAI)[2]. As a result, they use artificial intelligence to create more flexible and interactive systems that adapt "to the student's individual needs by analyzing and diagnosing his issues to offer him the appropriate aid" (Buche, 2005). The idea is to mimic the behavior of a human tutor in his role as a topic expert and expert instructor. This program may guide a learner through a task and offer pertinent comments like a teacher. ITS, therefore, reacts to the requirement to put the learner first.

As believed by Burns and Capps (1988), CAIs evolve toward ITSs, by bypassing these three intelligence challenges:

- First, an ideal system must understand the subject matter, or domain, for an embedded expert to make inferences or solve the domain's problems.
- Second, the system must deduce the approximation of such knowledge to the learner, in other words, the quantity of the knowledge that the learner has acquired.
- Third, the tutorial technique or pedagogy must be intelligent, so the ITS can use various strategies to narrow the gap between expert and learner performance.

As a result, three primary components, or models, treating one challenge each, compose the ITS operating functions. These Models reflect the three major components of any educational system: material to be taught, a way to assess the learner's knowledge, and intrinsic teaching or instructional style to increase the knowledge of that learner. These are known as Domain, Student, and Tutoring Models in ITS terminology (Clancey et al., 1979a).

According to Roberts (1984), the ultimate objective of ITSs is to create a system that combines powerful models in each of the three components, the Expertise Model, the Student Model, and the Tutoring Model.

1. **The Domain Model**, or the expert model, refers to the subject knowledge that the system imparts to the student. This knowledge involves both the stuff to teach and how to apply it to issues. This latter is known as procedural knowledge and represents the methods employed by experts to solve similar challenges. The domain model generates questions and evaluates the learner's answers.
2. **The Student Model** represents the learner's comprehension of the taught topic. So the system may point out the learner's misunderstandings and inefficient performance tactics, explain why they are wrong, and offer adjustments.
3. **The Tutoring Model** specifies how the system should display resources to the student. The tutoring model includes teaching methods and resources and

[2] In some literature works, they use Intelligent Computer-Assisted Instruction (ICAI) to refer to ITS

interacts with the student by assigning problems, monitoring and critiquing their progress, giving aid when needed, and recommending remedial materials.

Because most ITS programs are extensive and sophisticated, not all three components are fully developed in every system. Instead, most developed systems concentrate on developing a single component of a fully functional system(Clancey et al., 1979b).

1.2.3 Basic Models of Intelligent Tutoring Systems

Modern ITSs aim to imitate the work of an instructor or teaching assistant and progressively automate pedagogical responsibilities, including issue development, problem selection, and feedback production. In essence, ITSs are problem-solving or training environments. They facilitate learning in a particular domain by guiding and aiding the student. Sometimes they give the domain material first and then the activities that may assist the student digest the knowledge (Nkambou et al., 2010).

As a result, an educational system that can identify learning patterns and give personalized teaching instructions has been developed. As determined by most researchers, four fundamental components compose ITS models, the domain model, the student model, the Tutoring model, and the user interface model (Nkambou et al., 2010; Nwana, 1990) (Figure. 1.2).

1. The Domain/expert model has all the knowledge about the field of study (taught topic).
2. The Student model collects the student's data and uses them to create a representation of the student's knowledge state. This model also evaluates the student's level of knowledge by comparing the student's state of knowledge to that of the expert.
3. The Tutoring model determines which unmet needs to address and what tactics to use to communicate that information.
4. The User-Interface model allows for tutorial data to be transmitted (Burns and Capps, 1988).

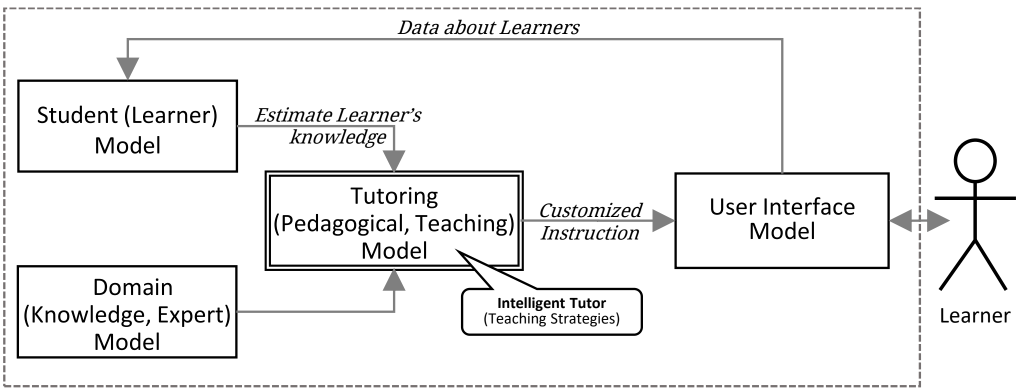


Figure 1.2 – *Basic Models of an Intelligent Tutoring System (Butz et al., 2006)*

1.2.3.1 The Domain/Expert Knowledge Model

The Domain Model, also known as the Cognitive Model or the Expert Knowledge Model, is the system's backbone, where most of the intelligence is stored. It includes the ideas, rules, problem-solving techniques, and student performance indicators needed to mimic expert knowledge processes (Orey and Nelson, 1993). The intelligence required for the expert model may necessitate substantial collaboration between the system designers and subject matter experts. As reported by several experts, at least some pedagogical information in every subject is inherently domain-dependent (Freedman et al., 2000). Some notions in astrophysics, for example, have no analogs in other fields. The more the domain independence, the more time and resources required to create future ITSs can be reduced (Fenza and Orciuoli, 2016).

1.2.3.2 The Student Model

The student or "learner model" provides teaching instructions by capturing the learner's knowledge profile. ITS uses the student model to identify the learner's mistakes, provide feedback, and provide level-appropriate questions to aid students in their development.

1.2.3.3 The Tutoring Model

The Tutoring Model, also known as a Pedagogical Model or a Teaching Model, keeps track of teaching classes and the student's activities. It uses information from the diagnostic process to establish tutoring techniques and actions (Orey and Nelson, 1993). It is layered on top of the domain and student model. When the learner deviates from the planned learning outcome, the model alerts the learner and gives rapid feedback to get them back on course. Learners can also request help. The tutoring model determines what information should be offered to the learner and when and how it should be presented.

1.2.3.4 The User Interface Model

The User Interface Model depicts the learner's intelligent engagement with the system; It is a communication channel that allows the learner to carry out the learning activity while the system interprets the activity and responds appropriately. Designers must predict various learner actions for a given interface to be effective. Learners may need to learn how to utilize the interface before acquiring the intended topic matter; therefore, an intuitive design is essential.

ITS projects vary significantly in scope and size, depending on the respective levels of intelligence of the four components (Freedman et al., 2000).

For example, a system that places a high emphasis on the domain model may produce a vast collection of complicated and unique issues for learners but still teach these problems to learners of various levels in a linear way. In contrast, an ITS that emphasizes several pedagogies to teach a topic may require a more engaging user interface but a smaller set of questions. The interface is prioritized in all ITSs development forms since it represents usability. The domain model should convert information to the interface, the student profile should always be captured (the student model), and the tutoring model should contain knowledge about how to offer instruction (Orey and Nelson, 1993).

1.2.4 Developped Intelligent Tutoring Systems

New computer-aided education paradigms, such as e-learning and distributed learning, offered an appropriate foundation for ITS concepts during the fast rise of the web boom. ITS has also been linked to or merged with other technologies like multimedia, object-oriented systems, modeling, simulation, and statistics. The success of ITS has historically affected non-technological fields such as educational sciences and psychology (Ramos et al., 2009).

While Intelligent tutoring systems developed from research in cognitive psychology and artificial intelligence, various applications are currently identified in education and business. Elementary school and college students benefit from intelligent tutoring programs, which may be obtained online or in a typical computer-lab classroom. Many systems target mathematics, but applications may be found in health sciences, language acquisition, and other fields of structured learning.

Reports of improved student understanding, engagement, attitude, motivation, and academic outcomes have led to continued interest in investing in and researching these systems. The adaptive nature of ITSs allows educators to develop customized curricula. There are several ITSs in education; a complete list does not exist, but a few more effective programs are included below:

- **ASP.NET-Tutor** An ITS that helps students learn ASP.net easily and smoothly (Mosa et al., 2018).
- **LISPP (Learn, Imaginate, Select, Practice, and Play)** System that helps college students overcome mathematical learning difficulties (Lafifi et al., 2019).
- **ARDUINO Tutor** An ITS for Training on ARDUINO. The system thoroughly introduces the Arduino platform while acting as a personal tutor, dealing with students in compliance with their abilities. Equipped with an open-source programming environment and library for creating code to control the Arduino board subject, the system adapts to the trainee's development (Albatish et al., 2018).
- **EER-Tutor** is a constraint-based tutor that teaches Entity Relationship design (Zakharov and Mitroví, 2004).
- **SQL-Tutor** a constraint-based tutor used to instruct students on databases' data retrieval using the SQL SELECT statement.
- **Practical Algebra Tutor (PAT)** modern algebraic tools were used to involve students in problem-solving and sharing their outcomes. PAT aims to foster progress by drawing on students' past knowledge and everyday experiences with mathematics.

1.2.5 Limitations of Intelligent Tutoring Systems

Even though the theoretical foundations of ITSs seem promising, they present major flaws and shortcomings that must be addressed to have more reliable systems. Among these limitations, we found:

1. One of the most common ITS system flaws is the student-computer conversation. Understanding natural language is a complex topic that the AI community is researching. Until then, current systems require students to utilize a subset of the language, generally with certain syntax restrictions.
2. A second flaw is the implicit assumption that we can grasp a student's knowledge by comparing the student model to the expertise model. We do not know how individuals think, so the expert model may not suit all students, especially given the cognitive-developmental phases that preadult students must undergo.
3. Third, the ITS system development is quite labor-demanding. Building an ITS system that teaches even a tiny amount of information takes a long time, frequently several person-years. Aside from research, the existing state-of-the-art prevents any ITS development.
4. A fourth flaw is the implementation of content domains. Mathematics, electronics, and gaming are among the most used ITS curriculum categories. The extensive application of ITS systems and models in various content fields has to be proven.

1.3 E-Learning Systems

Since the introduction of TVs and overhead projectors in classrooms decades ago, e-learning has evolved to incorporate interactive computer programs, 3D simulations, videos, telephone conferencing, and real-time online discussion groups with students from all over the world have become popular. E-learning progresses as technology advances, opening up a world of possibilities.

E-learning systems are innovative tools that combine Information and Communication Tools (ICTs) with social sciences to get their best. Introducing new forms of technology into traditional learning programs seems a good idea at first glance. However, such integration does not come without a price. This section opens a window on these technologies and their different sub-categories and similar technologies while covering their advantages and shortcomings.

1.3.1 What is E-Learning?

The term "e-learning" was initially used in a seminar held by "CBT Systems[3]" in October 1999. It was most commonly used to describe computer-based training

[3] CBT Systems, one of the founders of the Computer-Based Training industry, who changed its name to be "SmartForce." SmartForce Technologies, Inc. *https://smartforcetech.com* (Accessed: 10, May 2022)

delivered via intranets and the Internet. The word "E-Learning" replaced Web-Based Training, which was not attractive enough during the dot-com days. It was an era when everything would have an "e" in front of it: e-mail, e-letters, e-toys, e-commerce, e-banking, e-pets, and the list goes on. Placing the letter "e" in front of a verb indicates that a new Internet application was highly innovative at the time. Of all, "e" stands for "electronic," which is a far more commonplace and imprecise term than, Online or Digital, for example.

Over the years, the definition has evolved and included various variations where many experts offered different definitions of e-learning. However, there was a recurrent reference to offering classes through the Internet. Nowadays, e-learning implies different things to various people and sounds more casual as it is commonly used in our daily lives. It gained more popularity during the emergence of the Coronavirus and the Covid19 disease, where several countries were obliged to favor distance learning over traditional ones. Even though this choice was temporary and did not last much in some of these countries, we witnessed the rising social awareness of that learning method as a complementary means of education, allowing instructors to provide another layer of knowledge delivery.

The e-learning domain began with Computer-Based Training (CBT) implementation in a business context, which was considered a way to replicate the conventional educational environment. Initially delivered via CD-ROM, this early teaching method proved to be relatively static, making it impossible to track learners' progress. It was just a matter of time before e-learning expanded to include a media that would assure its global reach: the Internet. The Internet revolutionized the transmission of educational solutions in the same way it revolutionized information delivery.

E-Learning is defined by Brandon (2008) as "the use of technologies to create, distribute, and deliver valuable data, information, learning, and knowledge to improve on-the-job and organizational performance and individual development."

Based on the Cambridge dictionary[4], e-learning is an uncountable noun that means "learning done by studying at home using computers and courses provided on the internet."

As specified by the Oxford dictionary[5], e-learning is an uncountable noun that means "a system of learning that uses electronic media, typically over the internet."

The term "e-Learning" describes an educational environment in which teaching and learning take place within an Internet-based environment (Berge & Collins, 1995) and as "the use of digital technologies and media to deliver, support and enhance teaching, learning, assessment and evaluation" (LTSN, 2003, pg. 6).

For many people, the value of e-learning is not in its electronic character but in its ability to connect working, learning, and community in the workplace or institution (Mason and Rennie, 2006).

---

[4] E-Learning. in *Cambridge*
https://dictionary.cambridge.org/dictionary/english/e-learning (Accessed: 15, April 2022)

[5] E-Learning. in *Oxford*
https://www.oxfordlearnersdictionaries.com/definition/english/e-learning (Accessed: 15, April 2022)

1.3.2 E-Learning Terminology Confusion

As claimed by Mason and Rennie (2006), many e-learning terminologies are interchangeable, partly because various English-speaking nations have developed somewhat different applications and partly because the English language is developing as its use grows. The term "e-learning" is new, and numerous terms define the same activity. Before the Web, "Computer Conferencing" referred to the communications part of e-learning, while "telelearning" was popular in the 1990s but has since faded. The terms "Online Learning", "Virtual Classroom", and "Asynchronous Learning" are still used. These names generally relate to the same practice; nevertheless, some writers employ terminology that does not refer to the same behavior. "Virtual university" can refer to a university that offers online courses, but it can also refer to anything that is neither a university nor "virtually" online. Distance education is another phrase that might apply to correspondence education, videoconferencing, or even e-learning in other world areas.

(Mason and Rennie, 2006) explained those variations by how a person conceptualizes the area, namely which phrase serves as an umbrella notion that encompasses all other activities as subsets. For example, the term "flexible learning" is popular in Australia; and the "e-learning" term constitutes a subset of flexible learning as specified by the Australian Flexible Learning Framework Flexible Learning Advisory Group (2004):

> *"E-learning as a component of flexible learning describes a wide set of applications and processes which use any available electronic media in the pursuit of vocational education and training. It includes computer-based learning, web-based learning, virtual classrooms, and digital collaboration."*

However, referring to the American glossary, Learning Circuits[6], e-learning is a subcategory of Distance Education:

> *"The definition of distance education is broader than and entails the definition of e-learning."*

To decrease this confusion, we have to give more clarity about some educational terms and draw borders between them. In this study, we adopted and extended the classification of Mason and Rennie (2006), which shows how Distributed Education encompasses characteristics of Distance and Online Education and their relationship with Blended Learning. In Figure 1.3, we present a classification of the educational field based on two criteria:

1. Distance, or Remote education against Face-To-Face education.
2. Using traditional techniques against using ICT-based techniques.

In the following, we present a thorough definition of terms related to e-learning to relieve any kind of confusion.

[6] Learning Circuits American Glossary.
http://www.learningcircuits.org (Accessed: 16, February 2022)

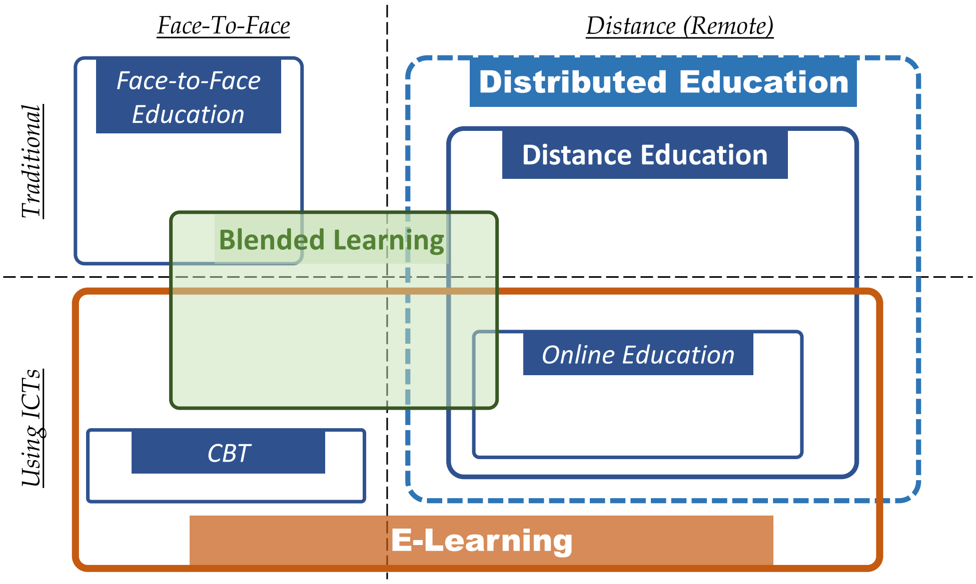


Figure 1.3 – *E-Learning in the Educational Field, based on (Mason and Rennie, 2006, p. xvii)*

1.3.2.1 Distributed vs. Face-to-Face Education

The term "Distance Education", sometimes referred to as Remote Education, appears to be self-explanatory, leading one to doubt the necessity for clarification. However, a survey of the literature reveals that distance education was not always precisely defined in the past; instead, the term was thrown around carelessly and was sometimes used interchangeably with expressions such as education by correspondence (Faibisoff and Willis, 1987).

Distance education is not a new concept. For over a century, educational institutions have reached out to learners who could not attend classes on a university campus due to distance, time constraints, or even professional duties. Learning materials in remote areas of the world were sought even before ICT technical advancements.

In distance education, face-to-face communication is impossible when instructors and students are physically separated. Instead, communication is conducted through one or more communication means, most usually but not always electronic, like television, computers, phones, or even ordinary mail. The distance between the instructor and the student might be significant, in an Internet-based course, for example, or minor, from the teacher's computer to the student's computer in a nearby campus building. As a result, the term "Distance Education" may describe both on-campus and off-campus courses and programs.

Distance education started as postal correspondence between the student and the educational institution. In the early twentieth century, instructional radio widened the reach of remote education. Television was adopted to reach a larger audience after the early success of instructional radio. With telecommunications and the Internet advances, distance education has reached a considerably greater audience than television, radio, and conventional mail. These strategies have expanded the scope of distance education. Schools now provide 100% online courses. Even though the Internet has become the most common distribution method for remote education, media such as television and radio continue to play an essential role in providing education to as many people as possible (Meere, 2012).

Zigerell (1984) defines Distance education as a form of instruction characterized by the physical separation of the teacher from the student, except for the occasional face-to-face meeting allowed by some projects.

Outlining the properties of the term may be the best method to define it (Faibisoff and Willis, 1987). There are many different meanings for distance education, but they all have essential qualities:

1. They allow students to contact instructors on occasion.
2. They allow for student independence and personalized study.
3. They are provided through on-campus and off-campus courses.
4. They are available based on the student's demands.

1.3.2.2 Online Education

In recent years, distance education has become one of the most discussed topics in education, with particular influences stemming from the "open" movement, such as MOOCs and OER. There have been numerous descriptions of online education and what it entails (Simonson et al., 2019). Distance education started in the 19th century as a correspondence study. It has had a significant impact on the way people learn. It has taken various forms and employed a wide range of technologies, from postal technologies in the 19th century to virtual reality in the present day (Saykılı, 2018). A new notion has emerged, "Online Education," to refer to distance education using ICTs

As alleged by Mason and Rennie (2006), Online Education is a subfield of Distance Education that uses ICTs, especially the Internet, to reach learners. These allegations were backed by Agostinelli (2019) who claimed that online education is the third generation of distance education, which has provided essential learning models and frameworks such as the Adolescent Community of Engagement and Community of Inquiry. Not far from these definitions, Larreamendy-Joerns and Leinhardt (2006) stated that online education is a new discipline that combines distance education with human-computer interaction, instructional technology, and cognitive science.

Nevertheless, online and distance education terms are used interchangeably in most literature, and some use distance online education.

1.3.2.3 Problems with Distance and Online Education

Unfortunately, the advancement of distance education technology has generated situations that are rarely, if ever, seen in academic life—conditions that pose fundamental problems regarding teaching and research norms. For example, the teacher does not have the same face-to-face interaction with the student as in regular classrooms. As a result, unique methods for assigning, mentoring, and assessing the student's work must be established. To communicate with the student, the instructor usually employs complex and expensive technology gadgets that are not under the teacher's sole control and frequently need particular technical skills that the teacher may lack. The syllabus, lectures, tests, and other course materials may be duplicated, recorded, and reused without the teacher's presence (Meere, 2012). In these new environments, the teacher's academic and legal rights may not be adequately understood or in dispute. Issues about the faculty's overall authority in selecting relevant rules and procedures for using these new technologies might also be a source of contention. Finally, the instructor should be compensated accordingly because the nature of teacher-student contact and the planning and delivery of distance education programs takes much more time than courses taught in typical classroom environments.

1.3.2.4 Blended Learning, an obvious solution?

The "Blended Learning" term has existed for a long time, and several definitions have been attempted.

As stated by Bersin (2004):

> *"The combination of different training 'media,' technologies, activities, and types of events to create an optimum training program for a specific audience. The term 'blended' means the traditional instructor-led training is being supplemented with other electronic formats."*

Aside from distance learning, Internet technologies have evolved to satisfy educational aims in both stand-alone and traditional educational environments. Hybrid or blended learning is a new approach to teaching and learning that combines traditional face-to-face meetings with online learning (Rogers, 2001). E-Learning is used alongside traditional teaching methods to promote accessibility and efficacy in these blended learning environments and create a richer learning experience. The lack of direct interaction, a primary component of many learners' learning experiences, is one of the negatives of complete online learning. The blended method also enables educational program architects to enrich existing programs by incorporating multiple media, allowing a level of mobility absent in traditional educational processes.

The low cost of e-learning in blended learning contexts benefits both learners and institutions. It is a cost-effective alternative to traditional teacher-led learning and is a great way to save money. Since many free-of-charge e-learning technologies and VLEs exist, the cost to implement and manage an e-learning system can be pretty low. Indeed, once created, e-learning materials can be freely accessed, with just a little cost of connection and update. Even with vast groups of learners, managing e-learning systems may be straightforward. Many Virtual Learning Environments (VLEs) have content management systems that allow teachers to control resource access and set learner access rates. Online e-learning sessions are incredibly straightforward to manage and update because learners can access updated content at their convenience. These technologies also let educators analyze learner progress and identify areas where learners may need more help. By engaging learners more actively, e-learning systems ultimately improve overall learning outcomes.

However, although Internet has enabled educational institutions to reach a far larger audience, due to the lack of social connection in solely online e-learning, it is critical that learners stay engaged and actively participate in sessions. Even though

ICT and the Internet provide media-richness and interaction to distance learners, social interaction and regular communication between communities are still needed to keep distance learners motivated (Meere, 2012).

1.3.3 Traditional, Online Education and Blended Learning

Traditional educational approaches used to be constrained by factors such as time and place. The knowledge transmission process took place in the classroom and was delivered to learners in a traditional face-to-face way by an instructor. However, as information and communication technology (ICT) and, in particular, Internet use grew in popularity in society, these technologies penetrated educational processes, specifically higher education. While e-learning was a very beneficial technology by itself, it still necessitated effective ways of incorporating it into educational processes. Educators were given the job of incorporating these technologies into existing teaching and learning procedures to fulfill escalating societal expectations. Several types of e-learning arose, each using an online component to differing degrees.

1.3.4 Why do so many people dislike E-Learning?

The issue is not with the technology itself but with its use instead. If e-learning is employed only to give incredibly boring, mandatory training, it is unsurprising that many receivers won't be unhappy; this technique would be unpopular regardless of how it was delivered. Some teachers are not convinced of the utility nor the efficiency of using e-learning instead of the traditional teaching that they are familiar with it. Add to that the fact that even if some of them want to give it a try, they do not have a clear image of how to do it, and even if they do, they lack the technical mastery to embody it. Another issue that could arise is excessive reliance on self-study. Although this is a very flexible method of delivering a learning intervention, it only works in tiny doses. We are social creatures that enjoy interacting with peers and experts.

1.4 Conclusion

ITSs arose from several fields like Artificial Intelligence, cognitive psychology, and education science. Historically, they focused on developing domain-specific research systems and, more specifically, on academic education. However, because it is primarily a research-driven field, implementations tend to be unique in the capabilities they provide, contain hand-crafted ontologies produced by small groups of developers, and lack compatibility with one another. On the other hand, E-Learning systems are enterprise-driven technology that has so far been primarily developed by higher education and workplace training institutions. Interoperability through standardization and wide-scale deployment are motivating factors for this community. Thus, traditional eLearning research focuses on reuse, component interoperability, integration with organizational software, and content authoring (Brooks et al., 2006).

The main features of ITS and eLearning systems are presented in Table 1.1.

ITSs	**e-Learning Systems**
Focused on improving learning	Organized learning and material presentation
Limited content	Huge content
Carefully crafted content	Content developed by regular authors
Single author/designer	Potentially collaborative writing
Abstract and fixed domain ontology	Several ontologies, based on the content
Elaborate feedback	Simple feedback
Some feedback generated	Predefined feedback
Tightly integrated components	Service-based approach
Few generalizable solutions	High scalability and reusability

Table 1.1 – *Comparison Between Typical ITSs and E-Learning Systems Features*

Many exceptions to this classification have emerged as efforts, like ours, are made to blur the lines between ITS and e-Learning.

At this point, one can claim that it is possible to make eLearning more intelligent and ITS more open and reusable while retaining their useful features. Web technologies, in particular, can be used to improve technological adaptivity, reuse interoperable components, and make systems more widely available and maintainable. ITS methodologies can be used to make adaptations that are genuinely beneficial to learning, such as student modeling, tutorial dialogues, and other valuable ideas and tools developed over time.

Online learning systems offer great tools and opportunities for learners and teachers alike. However, many theoretical and ideological concepts remain mere promises that have a long way to go, like the personalized and adaptive learning that of a unique learning experience for each learner following his profile and needs.

Online learning systems offer great tools and opportunities for learners and teachers alike. However, many theoretical and ideological concepts remain mere promises that have a long way to go, such as the personalized and adaptive learning that of a unique learning experience for each learner following his profile and needs.

We believe that meeting someone's needs includes knowing them first, then knowing how to satisfy them, and finally knowing the difficulties preventing them from being met. Finding and fighting these difficulties could greatly help enhance the teaching process. Learning difficulties faced by learners are not a new concern to deal with it. However, the lack of physical interaction between teachers and learners inside online and distance education environments makes the problem becomes deeper.

The next chapter dissects the issues and difficulties faced by learners in general and specifically in remote learning environments.

Chapter 2

Learning Difficulties: Diagnosis, Detection, and Prediction

Contents

2.1 Introduction 26
  2.1.1 *What are Learning Difficulties?* 27
  2.1.2 *Learners with Difficulties: "At-risk" and "Struggling" Learners* . 27
  2.1.3 *Learning Difficulties indicators* 28
2.2 Learning Difficulties Factors and Causes and Effects on the Learner 30
  2.2.1 *Teaching-Related Factors (Teaching Methods and Strategies)* . . 31
  2.2.2 *Curriculum-Related Factors (Domain Knowledge)* 33
  2.2.3 *Learner-Related Factors* 34
  2.2.4 *Classroom's Environment-Related Factors (Cognitive, Emotional, Behavioural, and Social)* 38
  2.2.5 *Importance of Early Identification of Learning Difficulties* . . 39
2.3 Identifying Learners with Difficulties inside E-Learning Environments 39
  2.3.1 *Detection of Learners with Difficulties using Artificial Intelligence* 40
  2.3.2 *Existing Detection, Prediction, and Intervention Systems* . . 40
  2.3.3 *Used Artificial Intelligence Techniques to Detect and Predict Learning Difficulties* 43
2.4 Conclusion 44

2.1 Introduction

People are different in how smart they are. Some people can quickly learn a wide range of complex skills, while others struggle to master relatively simple skills. People with learning difficulties may be found in all social categories in all parts of the world. However, it is difficult to estimate their numbers since it is dependent on persons being diagnosed or identified administratively. In the case of school-age groups, it is relatively easy to produce reliable estimations since learners with learning difficulties are frequently sent to other organizations such as psychologists, social services, or child guidance centers. On the other hand, many older people with mild to severe learning difficulties are no longer counted in statistics since they live independently (Hulme and Mackenzie, 2014).

As reported by Robinson (2002), learning difficulties begin in the first few years of formal schooling. That is also when these learners form important beliefs about themselves and their abilities. In some situations, they can start questioning their abilities thinking they are not good enough, especially when they cannot do something that others can. For example, if learners can not recognize words or work with numbers, this can significantly impact their confidence and motivation, pushing them to avoid dealing with failure-related activities. As a result, they avoid trying new or difficult situations, preventing them from improving their skills due to a lack of practice. From that point on, the downward path is set, and the cycle of failure starts.

Individuals with learning difficulties are, by definition, slower to acquire new skills and talents than the average population. Since the 1960s, interest in the unique abilities and cognitive processes that give these individuals particular difficulties has grown. Persons with learning difficulties are hindered from learning as they follow the same developmental and learning rules as ordinary individuals. Fortunately, with the new technological advances, it is possible to find these learners and assess their condition to help them. Furthermore, it is not uncommon to discover that some learners with academic difficulties also struggle with socialization, and others have difficulty adhering to social norms.

This chapter combed through the international literature to see what is known about learning difficulties and how schools and educators might best treat them. It focuses on the probable causes provoking or aggravating these difficulties and their emotional and cognitive effects on the learners. It also introduces the most pertinent and apparent learning difficulties signs and indicators that could be used to identify struggling learners.

This chapter also talks about early detection and prediction of these difficulties, so that intervention and support may be offered as soon as possible to prevent or reduce the harmful emotional effects of repeated failure. These negative consequences frequently serve to persist or aggravate a learning difficulty for the learners involved by lowering their self-esteem and decreasing their desire to study. It also looked into the literature to review conducted works and studies to diagnose, identify, predict and assist learners with difficulties within e-learning and ITSs environments. These works are grouped and presented in concordance with different criteria.

2.1.1 What are Learning Difficulties?

As mentioned by Westwood (2008), "learners with learning difficulties" is a broad phrase that is used carelessly and with little accuracy. Typically, the phrase refers to learners whose academic learning difficulties are not directly induced by any specific physical, sensory, or intellectual deficiency (although, in some cases, their intelligence may be somewhat below average). Instead, learning difficulties may be caused by external factors such as sociocultural disadvantages, inadequate learning opportunities, a lack of parental support, an improper curriculum, or insufficient early-childhood education. The emotional reactions of these learners toward failure frequently compound their learning difficulties. These learners have sometimes been labeled as "slow learners" and "poor achievers."

Badian (1996) refers to these learners as having "garden type" learning difficulties, implying that they are familiar and not unique. These learners are now commonly referred to as having general learning difficulties. Their failure can be seen in almost every aspect of the educational curriculum.

A far lower percentage of learners with learning difficulties may have specific learning disabilities (SpLD). Despite having an average IQ, these learners have persistent difficulties mastering fundamental reading, numeracy, and study abilities. They may also struggle to form healthy social connections (Westwood, 2008).

Karande et al. (2005) define specific learning disabilities (SpLDs) as a heterogeneous group of disorders characterized by significantly unexpected specific and persistent difficulties in the acquisition and use of efficient reading (dyslexia), writing (dysgraphia), or mathematical (dyscalculia) abilities despite conventional instruction, intact senses, average intelligence, adequate motivation, and adequate socio-cultural opportunities. The term SpLD does not cover learners with learning difficulties induced by visual, hearing, or movement impairments or below-average intellect.

2.1.2 Learners with Difficulties: "At-risk" and "Struggling" Learners

Westwood (2004) reports that various labels have been applied to students with learning difficulties over the years, including "learning handicapped," "dull," "educationally subnormal," "slow learners," "at-risk," "low achievers," and "hard to educate." Over time, each label accumulates rejection and is eventually replaced by another.

As believed by the same author, the descriptor "struggling," as in "struggling readers", appears to have gained popularity in American literature. It appears in the titles of several novels dealing with learning difficulties. However, it is hoped that the phrase "struggling" follows in the footsteps of other derogatory expressions, as it indicates that the student is at fault.

However, in the literature, we stumbled upon the term "at-risk" more often than "struggling," especially when speaking of identifying or predicting those learners. We do not have the exact statistics for using these terms; nevertheless, the former is more used determined by the Google Books Ngram Viewer [1] (Figure 2.1).

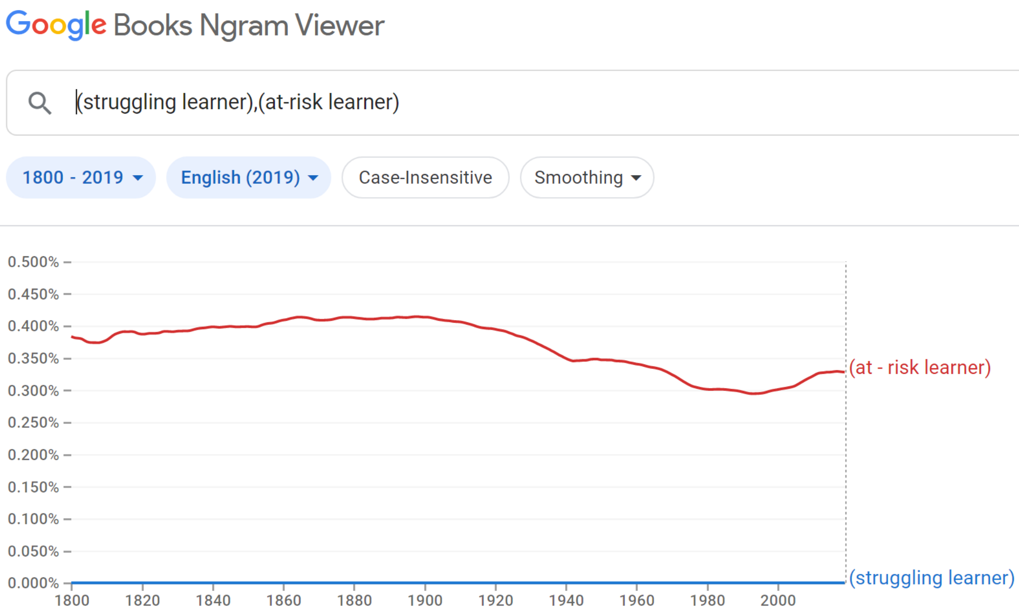


Figure 2.1 – *Usage of the terms "At-risk" and "Struggling" in the literature as reported by Google Books Ngram Viewer*

(Bernstein et al., 2013) defines a learner "in difficulty" as a learner who is "at risk" of receiving less than a "pass" in his following coming evaluations. Learners in difficulty are rarely aware of their situation for various reasons, including the lack of self-awareness or concern that they will be stigmatized if they acknowledge they are in difficulty. However, as alleged by the same author, early identification of such individuals with the appropriate intervention leads to better outcomes for these learners.

In the following, we will refer to learners having learning difficulties as "at-risk," "struggling," and "in difficulty" learners. To identify those learners, one may know the different signs and indicators that point the finger at such learners.

2.1.3 Learning Difficulties indicators

Typically, learners facing learning difficulties with fundamental arithmetic also have difficulties with reading. These literacy and numeracy issues impede learners' success in practically every subject. Moreover, learners appear to lack appropriate learning techniques for coping with the task provided by teachers, resulting in low accomplishment.

Many of these learners do not participate verbally during classes, do not show any interest in the subject matter, and do not see class conversations as learning opportunities. Their behavior acts as a defensive mechanism, shielding them from probable embarrassment if they provide an incorrect response, exposing their intellectual deficiencies (Twomey, 2006).

[1] Google Books Ngram Viewer: *https://books.google.com/ngrams*

From an information processing perspective, understanding the concept that these learners often have difficulty acquiring, interpreting, storing, modifying, and retrieving information is fundamental to understanding the learning challenges. In particular, they fail to spontaneously use learning techniques or previously learned information during certain specific cognitive tasks (Chan and Van Kraayenoord, 1998).

As reported by Kavale et al. (2006), there is no practical behavioral or academic checklist that could distinguish learners with general learning difficulties from learners with particular ones. Westwood (2008) identified and listed the most common problems as:

- low concentration on the activity and the teacher's instructions, resulting in a significant reduction of the time spent on active learning,
- disengagement,
- low self-esteem,
- dysfunctional attitude,
- negative behaviors,
- shortage in cognitive and metacognitive techniques to promote learning,
- memory and organizational problems,
- diminished self-efficacy,
- risk-taking avoidance and passivity,
- External locus of control and acquired helplessness,
- frustration,
- motivation loss,
- depressive tendencies.

2.1.3.1 Learning Difficulties models

(Twomey, 2006) proposes three viewpoints on learning difficulties and their underlying causes, each focused on distinct circumstances and different learner characteristics. All three models may be valid and not mutually exclusive. Learning failure affects a learner's self-esteem and confidence in all three models and leads to secondary emotional and motivational issues. These models are:

– **1. The deficit model** assumes that learning difficulties are caused by cognitive and perceptual deficiencies like Intelligence Quotient (IQ) below average, poor task attention, visual and auditory processing difficulties, memory issues, and inability to comprehend sophisticated instructional language. Additionally, disadvantages associated with the learner's cultural or familial background, such as a dysfunctional family environment, difficulties with English as a second language, low expectations, a lack of support, health problems, or poverty, may contribute to the deficit model.

– **2. The inefficient learner model** focuses on an individual's inability to address academic learning methodically rather than focusing on that individual's deficiencies. This perspective is more favorable because research shows that learners can be taught to become better learners.

– **3. The environmental factors model** holds that learning difficulties are mainly attributable to environmental factors, the most important of which is the quality and appropriateness of instruction, like the teaching techniques and curricula.

2.2 Learning Difficulties Factors and Probable Causes and their Effects on the Learner

As specified by Hulme and Mackenzie (2014), the causes of learning difficulties can be divided into three broad categories: congenital, environmental, and pathological. There is no single identifiable cause; in practice, many factors interact to put a learner in the face of difficulties:

– **Genetic and Environmental Variation:** Mental capacities vary widely across persons due to genetic differences and growing conditions. It can be measured by IQ testing and has a normal distribution due to the complex interaction of numerous environmental and genetic variables that determine the development of intellectual distinctions; This indicates that a tiny percentage of the population may always have IQs much above average. Conversely, there may always be a minority of persons with IQs much below the national average. In harmony with this hypothesis, most moderate cognitive difficulties occur from a mixture of hereditary and environmental variables such as inadequate nutrition, healthcare, and educational opportunities.

– **Congenital Abnormalities:** Aside from typical genetic differences that impact intellectual development, other congenital pathological diseases cause intellectual impairment.

– **Brain Damage:** Early brain damage is a common cause of learning difficulties. Such harm can occur for many reasons and phases of development.

We have to mention here that the second and the third type factors are beyond the scope of this study as they are related to medical conditions that can not be addressed in this context. As for the first kind of factor, we did not find a complete list of causes behind learning difficulties in the literature. However, Westwood (2004) proposes a list of possible causes that can contribute to learning difficulty:

- unsatisfactory or unsuitable education
- curriculum that is both irrelevant and inappropriate
- atmosphere in the classroom
- disadvantageous socio-economic condition
- student-teacher relationship is compromised
- infrequent presence at school

- health issues
- using a second language as a medium of instruction
- a lack of self-confidence
- emotional or behavior difficulties
- intellect is below average
- a sensory impairment

While it is true that some learning difficulties can be traced back to a learner's weaknesses, this is not the case for the great majority of students. The most common causes of difficulties are environmental variables, such as teaching methods and curriculum. Furthermore, Kershner (2000) states that for the majority of students, the problem lies in their experience and understanding rather than their inherent intellectual limitations or deficits.

As a result, we grouped learning difficulties causes into four categories, Teaching strategies, Curricula, the Learner himself, and even the Classroom Environment. These leading four components are addressed one by one in typical ITS architectures (Figure 2.2).

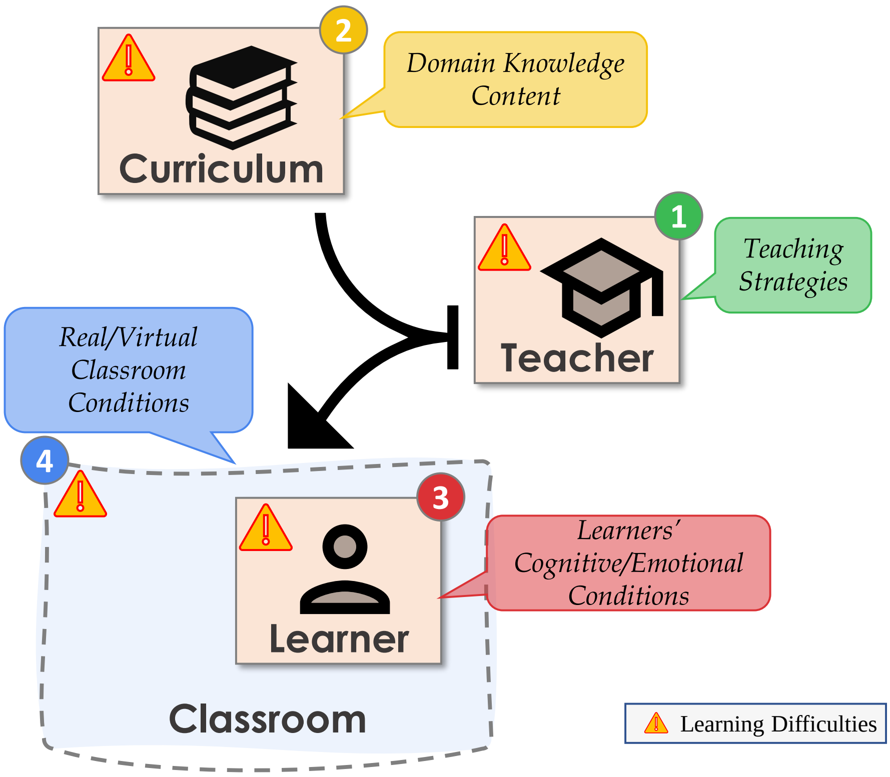


Figure 2.2 – *Learning Difficulties Factors and Probable Causes*

2.2.1 Teaching-Related Factors (Teaching Methods and Strategies)

Teachers still have a solid tendency to believe in the Deficit Model. They insist on blaming learners for their lack of enthusiasm or inadequate abilities. They rarely try to enhance the quality of their teaching or give learners advice on studying more effectively. If instructors perceive that learning difficulties are due to the learners' characteristics or external factors from family and society, there may be a general resistance to revising teaching techniques or materials. Unfortunately, instructors who believe in the deficit model typically lower their expectations of these learners, offering them a less-complicated and over-simplified curriculum that adds to their frustration and isolation leaving their basic age-appropriate need for success unmet (Westwood, 2013; Dettori and Ott, 2006; Elkins, 2007).

Teaching techniques and school curricula may frequently induce or worsen learning difficulties. Teaching methods were rarely explored as a probable source of learning difficulties until recently. Less effective teaching methods are also to blame for failures to maintain or increase the learner's ability, motivation, or persistence. Moreover, not all teaching approaches are equally efficient in accomplishing learning objectives, and not all strategies work well with all of them. Inappropriate learning methods cause issues, especially in the early stages of learning to read or compute. For example, if unstructured or learner-centered techniques are employed rather than direct teaching. Fortunately, nowadays, some instructors are feeling that many reading and arithmetic issues could result from improper or inadequate first-teaching (Westwood, 2006).

In the opinion of Dettori and Ott (2006), teachers tend to consider underachievers and learners with learning difficulties as a homogenous group with their typical peers as needing similar requirements. They do not make any additional provisions for them, and they also expect bad behavior from these learners in class, leading teachers to focus on classroom management rather than differentiated or modified education. Referring to the same authors, secondary school instructors, in particular, have a negative attitude toward learners with difficulties.

Some teachers prefer to blame students themselves or their socio-cultural and family backgrounds for the learning difficulties. If willing, teachers have the most power to influence the learning environment because of how they educate. Learning difficulties are likely to occur when methods are not suited to students' interests and skills and when they are not adequately matched to the type of learning engaged in the lesson. For example, Cheng (1998) and Westwood (1995) found that teachers often blame students' weaknesses or impairments rather than their teaching method, curriculum, or relationship with the teacher. They talk about students being "slow", deficient in intelligence, disorganized, and poorly motivated. They also identify them commonly as coming from "poor home backgrounds" or "unsupportive families". Henderson (2002) refers to this as the "deficit model" surrounding learning difficulty, and Cheng (1998) and Westwood (1995) believe that this "blame the victim" approach can negatively impact teachers' classroom practices and the expectations they hold for students with difficulties.

Another teaching method issue is when the teacher moves too quickly in the program, giving less time to practice or use complex language when explaining. Sometimes, the teacher lacks appropriate teaching materials such as books or computer programs. Furthermore, some teachers distract the classrooms with too many activities at the same time (Abosi, 2007; Benyounes et al., 2020).

Problems emerge when the teacher becomes uninformed of his learners' needs because he did not or could not closely monitor them. If a learning issue is not identified and addressed soon, it may worsen. Sometimes learners with different abilities may be given tasks that are too complicated or too simple; both situations can lead to frustration and disengagement (Westwood, 2006).

In his book, Westwood (2006) investigated in depth the many various teaching styles, explaining their possible benefits and drawbacks. That author specifically highlighted instructional practices that may lead learners to have learning challenges, either directly or indirectly. He believes that many examples of learning difficulties may be traced back to unsuitable or insufficient instruction rather than faults in the learner.

2.2.2 Curriculum-Related Factors (Domain Knowledge)

Robertson et al. (1994) state that teaching methods are not always to blame for contributing to learning difficulties, but so can curriculum content.

In some cases, the curriculum itself can be a source of learning difficulties, especially if the material is too complex and beyond the cognitive abilities of some learners or if the assignments and activities are boring, causing learners to lose interest and attention. When learning activities in a curriculum are too hard or too easy, learners may lose interest in learning. An ideal curriculum should be challenging enough to keep the learners' curiosity but not so tricky that it confuses or discourages some. It does not matter how hard one tries if the difficulty of the material increases too quickly (Paas et al., 2004).

The phrase 'curriculum impaired' was coined by Elliott and Garnett (1994) to describe a situation in which some learners are unable to cope with the subject matter's cognitive demands or the rate at which new concepts and skills are taught. When learners are given work beyond their abilities, they would be unable to connect the new learning to prior knowledge. As a result, learning becomes fragmented, and forgetting happens quickly. Student's self-esteem and motivation are severely harmed when they are confronted with painfully complex tasks day after day (Leiding, 2002) Another issue resulting in many, or most, learning difficulties, as explained by Howe (1999), is caused by learners lacking the requisite prior level of knowledge or expertise for the job at hand, rather than by cognitive deficiencies in the learner.

As stated by Van Kraayenoord and Elkins (1998), learners will soon become disengaged if the classroom instruction does not link with their lives and if it does not engage them as learners with themes and concerns that have interest and meaning for them.

Brennan (1985) proposed that curriculum content destined for learners with learning difficulties should be chosen to be Real, Relevant, Realistic, and Rational.

- **Real:** means that the curriculum should contain themes relevant to the learner's life and could be taught in concrete or experiential ways.
- **Relevant:** indicates that the built-in knowledge, skills, strategies, and assets are helpful to the learner when learning the topic.
- **Realistic:** means that the learner can successfully (Realistically) accomplish the task considering his age, ability, motivation, and prior knowledge.

- **Rational:** suggests that the learner recognizes the importance and purpose of participating in this learning.

Unfortunately, many school curricula do not meet Brennan's requirements, which leads to or aggravate learning difficulties in most cases.

2.2.3 Learner-Related Factors

Even though learning difficulties could be provoked or escalated by factors such as poor or limited teaching or inadequate curriculum content, some other factors are related to the learners themselves and could predispose them to learning difficulties or emphasize them. The learners' related factors can be either intrinsic or extrinsic. Extrinsic factors exist outside the learners' minds who encounter them, like health or family issues or socio-economic problems and the absence of support at home. In contrast, intrinsic factors reside within the learner himself like poor intelligence, Sensory impairment, developmental troubles, regular absences, cognitive abilities, perceptual disability, learning impairment, deficiencies in attention and memory, or lack of motivation (Farkota, 2005).

Although many extrinsic factors, such as health or family background, or income, are complicated and difficult to change, intrinsic factors such as the lack of motivation, external locus of control, or self-esteem could be addressed to suppress the negative emotions triggered by learners' successive failures (Westwood, 2006).

Wearmouth (2002) argues that for anyone to succeed in a given learning activity and keep putting up adequate effort, they must have faith in their abilities. Learners who have been unsuccessful in school for long periods begin to assume that they "lack talent and will never succeed". As a result, they abandon any honest attempt to achieve their assigned schoolwork.

As reported by Westwood (2008), the key emotional elements connected to a learner's success within the learning process include:

- **Intrinsic Motivation** how highly a student values a certain learning assignment.
- **Self-Esteem** valuing one's positive characteristics and personal value
- **Self-Efficacy** self-confidence in completing the assignment.
- **Self-Worth** Be aware of how others may see you as a learner.
- **Locus Of Control** ideas on the reasons behind success or failure.

2.2.3.1 Interinsic Motivation

Teachers frequently attribute a student's learning difficulties to a lack of motivation. It is almost as if teachers assume that motivation is a constant and possessed attribute of learners rather than a variable characteristic heavily influenced by external factors. For several learners with difficulties, their problem is undoubtedly not associated with a lack of motivation but rather a significant unwillingness to take chances or make any new commitment in a learning scenario. This hesitancy is related solely to earlier experiences of failure. Difficulties in learning impair a learner's motivation significantly because it is hard to sustain strong interest and devote enormous effort to learning anything if the outcome is disappointing.

Westwood (2008) believes that motivation is weakened by:

1. meaningless or dull tasks
2. cognitive overload
3. lack of flexibility in teaching methodology
4. unpleasant stimulus and criticism
5. lack of success.

Improving motivation to learn is an essential aspect of overcoming learning difficulties. While intrinsic motivation, that is, attempting a task out of genuine interest and value is desirable, extrinsic motivation, in the form of rewards and remunerations, is more likely in the early phases of dealing with learning difficulties.

2.2.3.2 Self-Esteem

The term self-esteem is strongly connected to conceptions of self-concept and self-image (Santrock, 2006). Positive self-esteem is required for maximum mental health, and it is the responsibility of all academic institutions to help learners build positive self-esteem. Self-esteem also determines one's motivation to attempt particular things and meet specific challenges. Ormrod (2005) emphasizes that teachers need to respond to learners' efforts in ways that may raise rather than undermine their self-esteem.

Positive self-esteem develops as a result of success. Low self-esteem is caused by a lack of success in a learning scenario. All learners must have several opportunities to succeed in academic, social, and physical settings to develop positive self-esteem and retain high motivation levels. It is critical in the academic sphere to tailor schooling to learners' developmental stages and capacities and give them both positive and constructive feedback. As claimed by Seligman (1995), self-esteem is virtually totally determined by an individual's accomplishments and failures in the world. Self-esteem grows due to successfully facing challenges, working hard, and conquering barriers.

Self-esteem is fragile in many aspects. It is challenging to reconstruct after it has been damaged. However, one of the main goals of working with individuals with learning difficulties is accomplishing just that.

2.2.3.3 Self-Efficacy

Self-efficacy refers to a person's awareness of his competency in a specific scenario. The individual's awareness grows over time due to observing his performance and the results obtained in various scenarios. Acquiring good results, being recognized and liked by others, having success, and knowing you're doing well all contribute to forming positive self-efficacy beliefs. Contrarily, poor results and excessive criticism impair a learner's self-efficacy and goals (Gage and Berliner, 1998; Biggs, 1995).

Achievement and self-efficacy are strongly linked. Knowing that you are performing well boosts one's emotions of competence and confidence, and the opposite is also true. Self-efficacy is a critical factor in determining how much effort learners will put into a task and how long they will persevere if the activity is complex (Moriarty et al., 1995). Lancaster2005 states that self-efficacy beliefs contribute significantly to the degree and quality of human functioning as they influence how people feel, think, motivate themselves, and behave. When faced with a new problem, a learner's expectations for success are strongly tied to his self-efficacy beliefs.

Learners with difficulties have been observed to have low self-efficacy, especially in academics (Klassen and Lynch, 2007; Lancaster, 2005). Individuals with poor self-efficacy avoid challenging jobs that they perceive as personally hazardous because they fear failure and losing face in front of their peers. They tend to have very negative beliefs about their self-efficacy due to a history of poor results from their efforts (Ormrod, 2005).

Providing hard yet feasible assignments, alongside teachers using descriptive praise while providing feedback, are critical in this matter. Verbose appreciation explains why a specific effort result is worthy of admiration. When learners believe that verbose admiration is sincere and credible, it boosts their motivation and self-efficacy.

2.2.3.4 Self-Worth

Self-worth is tightly related to self-esteem and self-efficacy because all three are involved with how people feel about themselves, When it comes to learning difficulties, self-worth directly impacts how some learners respond to challenges and potential failure scenarios. Self-worth theory focuses on how we attempt to protect ourselves from other people's unfavorable judgments (Eccles et al., 1998). Many learners with learning difficulties, for example, do not want their peers or teachers to believe they lack skill in a specific subject, so they create the appearance that when they receive poor grades, they have not tried hard enough. For them, it is preferable to be accused of not trying than to try hard and fail. In compliance with self-worth theory, a student may profit more by not doing the activity because avoiding failure would result in a loss of dignity. Avoidance protects the student's sense of self-worth in this situation (Valas, 2001). To retain self-worth, a learner may engage in defensive and avoidance behaviors, common among learners with learning difficulties.

A teacher is accountable for increasing all learners' feelings of worth by respecting their contributions, expressing interest in them as individuals, and employing the self-esteem and self-efficacy tactics mentioned above. Teachers must also encourage learners who exhibit defensive and avoidant behaviors to complete all of the assigned activities in class and provide any required help to ensure their success. It is critical that teachers publicly recognize and applaud their students' positive efforts rather than emphasize their learners' lack of effort.

It's generally helpful to help a student with learning difficulties evaluate his feelings, beliefs, and attitudes about the difficulty as a first step. Then, the student can utilize positive self-talk to overcome personal hesitation and restore some sense of self-efficacy. Counseling is frequently a required component of assistance.

2.2.3.5 Locus Of Control

Low self-efficacy confidence is frequently accompanied by what psychologists call an external locus of control. To grasp the notion of locus of control, one must first understand that people ascribe what happens to them in a given scenario to either internal or external variables, like their skills, talents, effort, or actions. Learners who have an internal locus of control understand that their actions can impact events, and they believe that they have some control over what occurs to them. Feelings of self-efficacy include recognizing that results are within one's control (Pajares & Urdan, 2006). Part of the action can be seen in the classroom when learners know that they obtain far better results if they pay attention, ask questions, concentrate, and work diligently. Learners may be less motivated to learn if they attribute their success or failure to forces above them that they have no control over.

Usually, as one gets older, one's ability to control their actions steadily improves. However, many learners with difficulties have a strong external locus of control when it comes to school learning, especially after horrible school experiences. They feel that their efforts in school have little impact on their success and that what occurs to them has nothing to do with what they do (McCowen, 1998; Bender, 2004). If learners suffer early mistakes and frustrations, their positive belief in their skills quickly diminishes. Previous inferences about a learner's achievements and failures are significant predictors of future motivation and achievement. A learner who keeps his locus of control mostly external is more likely to be managed and controlled by others, such as a teacher, assistant, or more capable peers.

The teacher's job is to let the learner realize how much power he has over events and how much he can impact outcomes (Galbraith and Alexander, 2005). It is natural for a teacher to want to assist and support a learner who has learning difficulties, but this should not be done to the point where all obstacles and difficulties of failing are removed (Fox, 2003).

Failure must be possible, and when it does, learners must be assisted in understanding the causal relationship between their efforts and the outcomes. Accepting occasional failure and assigning it to the actual cause is critical to learning (Seligman, 1995). Learners' judgments of incompetence may lessen when they realize that their mistakes are frequently caused by a lack of effort or attention. When students see that effort and persistence can overcome failure, they become more internal in their locus of control and more engaged in learning tasks. Self-efficacy is one of the most critical internal motivational tools for learners (Lancaster, 2005). A learner's willingness to persevere in the face of a tough assignment can be harmed by an external locus of control.

When the likelihood of failure is great, it is easier for learners to give up and develop avoidance methods rather than persevere. When a task is done, learners are taught to carefully appraise the results of their efforts using the instructional strategy known as attribution retraining (McInerney and McInerney, 2006).

2.2.4 Classroom's Environment-Related Factors (Cognitive, Emotional, Behavioural, and Social)

The classroom's physical environment might increase learning issues. The noise level and many distractions can affect some learners' attention span and on-task behavior. Temperature, lighting, the presence or lack of exciting display material, resources, and workspace on desktops can all help or hinder the learning process. For easily-distracted learners or hearing impairments, background noise sources can be particularly annoying (DuPaul, G.J. & Stoner, 2003; Alban-Metcalfe and Alban-Metcalfe, 2001).

The way learners are grouped and seated during classes impacts time on task, motivation, and involvement. For example, research shows that learners are more productive and on-task when seated in rows rather than in unstructured groupings (Hastings and Schwieso, 1995). However, many constructivist classrooms have learners seated in groups to promote cooperative learning and debate. A class of four or five learners seated around a table may be working in groups, but the learners are working alone. This potentially distracting structure can lead to lower achievement since learners struggle to focus (Lyle, 1996). A group environment is troublesome for learners with behavioral or attentional issues (Jenkins et al., 2003).

As reported by Goldberg (2002), class size is one aspect of the learning environment that has been intensively examined, with literally hundreds of studies conducted.

The smaller the class, the better for pupils' learning. Smaller classrooms allow teachers to recognize learners' needs better, provide individual attention, and adapt education to student variances. Studies reveal a far more nuanced picture. Reducing class size does not inevitably increase achievement or decrease failure rates because other factors, including student behavior and teaching quality, are crucial. Teachers may still teach small classes the same way they taught large ones, with no adjustments and limited individual attention for learners. The teacher is forced to employ a structured approach and maintain strong classroom management, resulting in higher overall accomplishment. However, enough studies have shown clear benefits from more minor courses to draw three tentative conclusions (Finn and Wong, 2002; Wilson, 2006):

- When classes have less than 20 learners, class size begins to have a favorable influence.
- The advantages are particularly noticeable during the early years of schooling (age 5 to 8 years).
- Smaller classrooms appear to assist learners from low-income families and learners in other high-risk groups the most.

Low self-esteem, low self-efficacy, low self-worth, and an external locus of control are all symptoms of learned helplessness. Learned helplessness appears to be more common among learners with learning difficulties who have low academic accomplishment. They begin to believe that everything a teacher wants them to accomplish will be too difficult for them and that they will fail, which may provide a significant barrier to future learning (Valas, 2001).

2.2.5 Importance of Early Identification of Learning Difficulties

The long-term impact of a learning difficulty can be devastating for the learners concerned, causing low achievement in key curriculum areas and may stimulate the affective consequences discussed in the previous section. Therefore, it is vital to identify learners at risk of possible learning failure as early as possible to provide appropriate assistance to minimize the impact of a learning difficulty. There is evidence to support the view that early intervention for problems in learning in both literacy and mathematics can have highly beneficial outcomes in terms of higher success rates in school and a reduction in the emotional problems associated with failure (Campbell and Ramey, 1994; Dowker, 2004; Wright, 2003; Siegel and Brayne, 2005; Wasik, B. A., & Karweit, 1994)

While identifying intellectual, physical, or sensory disabilities reasonably early in a child's life, the identification of less obvious difficulties in learning that are not related to a disability typically does not occur until he is in school and already having problems (Milton, 2000). This is sometimes referred to as the "wait-to-fail" model of identification and is regarded as less than satisfactory. Many attempts have been made to identify specific signs within preschool children's developmental patterns, behavior, or overall performance that might predict later learning difficulties. The main approaches to this problem include screening procedures and structured observation by teachers (Sugai and Evans, 1997; Twaddell, 2001; Leung et al., 2007).

The present section discussed learning difficulties within traditional education, how struggling and at-risk learners can be identified, and what measures are taken to help this category of learners. The following section will shed more light on the detection and prediction of learning difficulties and at-risk learners within artificial learning environments using ICTs and how early intervention can be performed to save the day.

2.3 Identifying Learners with Difficulties inside E-Learning Environments

In their quest to enhance the teaching process, e-learning platforms have gained popularity over the past few years, and for many good reasons. Learning software has come a long way in providing a comprehensive, fun, interactive, and engaging learning experience. Many governments, through their academic organizations, especially universities, turned their interest toward that kind of learning and all the luxury that came with it.

Nevertheless, these remote environments have some drawbacks, like the massive number of dropout learners, which can indicate the presence of some learning difficulties that can affect the quality and the quantity of the knowledge acquired by those learners and can, at worse, push them to withdraw the whole program.

Even though these difficulties are present in traditional education, their reasons can be utterly different in e-learning environments. For example, issues of isolation, disconnectedness, and lack of technical mastery may be factors that can increase the degree and number of faced difficulties faced by remote learning students and could drive them to leave online courses (Willging and Johnson, 2004). To understand this phenomenon, we started by exploring the literature to determine why learners drop out of their studies; Then, we extended our research to find the best indicators to perform an efficient prediction. In the end, we studied existing solutions for fighting this phenomenon in distance learning environments.

It is important to note that detecting and predicting struggling learners are two separate matters that rely on distinct factors and indicators. The detection process focuses on identifying learners who are having real-time difficulties. This procedure takes place “right now” by examining factual data to declare, "This learner is having difficulties right now." The Prediction process, on the other hand, is a procedure that seeks to forecast how learners will perform in the future or, even better, which of them may have difficulties. A good prediction could provide enough time to intervene early to support targeted learners.

2.3.1 Detection of Learners with Difficulties using Artificial Intelligence

The majority of students have a simple goal, to graduate with a degree that proves their knowledge. Artificial Intelligence (AI) can assist students in meeting this objective by streamlining the education process. By allowing access to the proper courses, improving communication with teachers, and freeing up more time to focus on other aspects of life, AI can have a noticeable impact on the student’s educational journey.

While it is not uncommon for learners to require additional assistance outside of the classroom, many teachers do not have the time to assist them outside school. Most instructors and teachers are not afraid to acknowledge they have time management issues, which is expected considering the number of tasks on their daily to-do lists. Educators desire to spend more time instructing students one-on-one, doing research, and furthering their education, but they do not have the time. By automating activities, assessing student performance, and eliminating the educational gap, AI can help free up instructors’ time. In all these situations, AI instructors and chatbots can be ideal.

While no chatbot can truly replace a teacher, AI systems can help students develop their skills and strengthen their weak spots outside of the classroom. They provide a one-on-one learning experience without the teacher being present at all day hours to answer queries. AI can provide teachers with a clear picture of which courses and lessons require reevaluation by analyzing students’ learning capacities and histories. Teachers can utilize this data to create the best possible learning pathway for all of their students. By analyzing each student’s unique needs, teachers and instructors can restructure their courses to address the most prominent knowledge gaps or issue areas before a student falls too far behind.

2.3.2 Existing Detection, Prediction, and Intervention Systems

Many developed learning systems have considered means and techniques to detect and measure, at best, difficulties faced by enrolled learners. Each reviewed research has a specific purpose and uses different indicators. To understand the relationship between Learning-Difficulties-related works, we have grouped them following their respective “Purpose of Research” objectives and the “Used Indicators”.

2.3.2.1 Purpose of Research Works

In Table. 2.1, we used the "Purpose of Research" as a criterion for grouping the found scientific works covering the topic of struggling learners. For each work, we have listed their used indicators. Results show that 36% of the works focused on identifying learners who are or would be "struggling." In contrast, 29% of the works tried to predict the outcomes of the learners rather than simply identifying the struggling ones themselves. In addition, 15% of the selected works aim to analyze and visualize the learners' current status in real-time for both learners and instructors. Systems offering that kind of functionality are sometimes called "dashboards."

2.3.2.2 Used Indicators

Even though scientists tried to achieve many goals, they almost relied on similar predictors. We first analyzed, then grouped research works focusing on the detection or the prediction of struggling learners to see which ones were the most significant indicators or factors used to identify "struggling" or learners. We grouped the works into two categories: performance-based and behavioral-based indicators.

In the first group, we found the work of Sandoval and his co-authors, who stated that the "students' grade point average (GPA)" was the most relevant indicator among 36 other indicators, followed by "the school in which the students were enrolled" as a moderately relevant indicator (Sandoval et al., 2018). In the work of Howard et al. (2018), the authors affirm that "continuous assessment" is the best indicator among three categories: "students' background information," "students' engagement," and "continuous assessment results."

In the other group, we found the work of Aggarwal and his co-authors. They found that demographic indicators such as "age," "gender," "location," and "family income" are more significant in predicting the student's outcome at an early stage (Aggarwal et al., 2021). Furthermore, Tarimo et al. (2016) affirm that "engagement" and "learning speed" are the best indicators for the final course grades. You (2016) and his co-authors have identified the "regular study" as the most pertinent indicator of the learners' performance, same as "submission delay" and "proof of reading" of the course (You, 2016).

Extensive research has been conducted to determine the factors that correlate to identifying or predicting learners' difficulties and failures in a course. Weak academic outcomes are a strong indicator of academic difficulties, thus suggesting a greater likelihood of dropping out. Learners who demonstrate poor educational strategies and lack persistence in reaching their objectives in life have weak academic performance and unsatisfactory completion rates, leading to a substantial risk of dropout (Pierrakeas et al., 2020). In their quest to find the best predictors, some researchers compared a set of academic and non-academic predictors to determine the best ones. Among found works, we summarized the most pertinent ones in Table 2.2.

We can divide the predictors into two categories, previous academic achievement and the learner's behavior inside the system. In the first category, information about past learners' performance could be beneficial to profile them if available. Unfortunately, this information could not always be found or accessed, even in academic organizations, and it is absent in non-academic ones, making these predictors

Research Purpose	Works	Used Indicators
Identify or predict struggling learners	Von Hippel and Hofflinger (2021)	Performance
	Mubarak et al. (2020)	Engagement, Time Series, Persistence
	Sarra et al. (2019)	Performance, Motivation and Resilience
	Casey and Azcona (2017)	Concept reuse
	Kuzilek et al. (2015)	Demographic, Traces
	Jayaprakash et al. (2014)	Demographic, Performance, Assessment
	Yukselturk et al. (2014)	Demographic, Self-Efficacy, Readiness, Locus of Control, Previous experience
	Lykourentzou et al. (2009)	Demographic, Performance, Assessment
Predict learner outcomes	Aggarwal et al. (2021)	Academic History, Demographics, Financial
	Baneres et al. (2019)	Performance
	Denden et al. (2019)	Performance-based Behavior
	Howard et al. (2018)	Demographic Information, Performance, Assessment
	Sandoval et al. (2018)	Demographic, Performance, Assessment
	Waddington et al. (2016)	Performance, Academic History
	AL-Malaise et al. (2014)	Performance, Academic History
Analyze and visualize learners' behavior	Kokoç and Altun (2021)	Interaction with Dashboard, Performance
	Azcona and Casey (2015)	Traces
	Papanikolaou (2015)	Cognitive and Social indicators
Describe dropout learners' behavior	Lakhal and Khechine (2021)	Technological, Demographic , Social, Psychological
	Zhou et al. (2020)	Academic History, Interaction Traces, Time Series, Financial
Compare predictive models	Maldonado et al. (2021)	Profit Metrics
	Marbouti et al. (2016)	Traces
Measure learner engagement	Toti et al. (2021)	Performance, Behavioral
	Hussain et al. (2018)	Traces
	Rajabalee et al. (2020)	Performance, Behavior
	Deng et al. (2020)	behavior, Performance, Emotion and social engagement

Table 2.1 – *Dropout-related works grouped by "Purpose of Research"*

less reliable and non-generalizable. As for the second category, monitoring and understanding the learner's behavior inside learning environments to predict his fate is tempting if successfully implemented. However, the human being is complicated and very hard to understand, making the use of this kind of predictor less precise. Nevertheless, dividing this category into two subcategories, the "learner's current performance" and the "time series," could simplify things. As for the learner's

Work	Goal	Used Variables (predictors)	Best predictors
Von Hippel and Hofflinger (2021)	-Identify students at risk of dropout	-Past Academic Performance (PAP) (college or University Grade Point Average(GPA)) -Entrance exam score (PSU) -Demographics (Sex, Family Income, Age)	-PAP -PSU
Aggarwal et al. (2021)	-Predict the student current performance	-Previous Academic Performance (PAP) -Demographics(Age, Gender, Income)	-PAP -Demographics
Kokoç and Altun (2021)	-Predict the student outcome	-Behavior (Time Series)	-Behavior
Lakhal and Khechine (2021)	-Determine the influence of technological factors on persistence	-Performance expectancy -Effort Expectancy -Social Influence -Facilitating Conditions, -Attitude towards using LMS -Anxiety towards using LMS -Demographics (gender, age) -Prior Online Course Experience	-Emotion -Performance
Howard et al. (2018)	-Predict learner's final mark	-Demographic (students' background information) -Continuous assessment; -Behavior (Interaction with the system)	-Continuous assessment
Sarra et al. (2019)	-Identify learners who are about to drop out	-Demographic (Family educational and socio-economic class) -Performance (Current Academic Competencies) -Orientation and choice of study -Intrinsic Motivation -Social and academic integration	-Performance -Motivation

Table 2.2 – *Comparative table of the best dropout predictors*

performance, it could be tracked based on his different evaluations and assessments. The access-times series tracking could be performed by monitoring his timeline.

2.3.3 Used Artificial Intelligence Techniques to Detect and Predict Learning Difficulties

Among existing AI techniques, the majority of these systems found in the literature chose to use Educational Data Mining (EDM), Learning Analytics (LA), and Machine Learning (ML) techniques. In the real-time detection of learning difficulties, the majority of the works chose to use LA. However, regarding the prediction, most of the found works have chosen to rely on Machine Learning (ML) to perform their predictions.

Educational data mining (EDM) used to be the most used analysis technique in education. However, the arrival of machine learning (ML) methods has changed that equation. Novel ML-based studies are now addressing the problem of analyzing and interpreting the behavior of learners to understand the dropout phenomenon. Many mathematical styles were used to analyze the traces of online learners (Mubarak et al., 2020).

2.3.3.1 Learning Difficulties Detection

In the first category (using LA), we found "iMoodle", an Early Warning System (EWS) that can predict struggling students and then intervene to save them Denden et al. (2019). "INSPIREus" is another system capable of analyzing and visualizing some specific indicators in the form of detailed views within an adaptive educational hypermedia system Papanikolaou (2015). In that direction, Azcona and Casey (2015) have designed a Virtual Classroom Environment (VLE) for learning the Assembly programming language. The proposed VLE is a real-time dashboard that can provide teachers with instant feedback on their learners' successful and unsuccessful compilations. Finally, "Course Signals" is a web-based system that provides real-time feedback to students while measuring their engagement based on their interactions with the Purdue Learning Management System called "Blackboard Vista" Jayaprakash et al. (2014); Pistilli and Arnold (2010).

2.3.3.2 Learning Difficulties Prediction

In the literature, we found that the most used ML algorithm to predict Learning Difficulties are Decision Tree (DT), Neural Networks (NN), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) (Table 2.3).

We summarized all other ML techniques used to predict the dropout learners or their outcomes in Table 2.4.

2.4 Conclusion

This chapter treats the problem of learners with learning difficulties. The first part addressed learning difficulties in the traditional education system from a psychiatric point of view, while the second part presents how ICTs and especially AI were used to diagnose, identify and predict those difficulties within e-learning environments.

It states that both performance-based and behavior-based indicators are good indicators. Each of them provides a significant amount of valuable information that can be used to report the learner's situation, suggesting combining them both to closely monitor the learners' behavior while performing their online activities. This behavior analysis can better perceive the learners' status, such as their punctuality, persistence, or even readiness.

It also asserts that no one can deny the benefits of using Machine Learning algorithms in the automation processes, especially its ability to improve with each new cycle, getting more accurate and precise results, a widely recommended quality in the prediction domain. However, ML algorithms require massive datasets and much training time to achieve that level of excellence. Any lack of such exigent conditions could affect its quality and make ML techniques a lousy choice in cases where there are few amounts of data or where the prediction time is vital. Moreover, the nature of the detection or the prediction operations requires monitoring all the actors' interactions individually, which requires the use of distributed and autonomous intelligence to obtain reliable results in a brief time. In most of the studied works, the only kind of artificial intelligence presented is centralized artificial intelligence like Machine

Machine Learning Technique	**Variant**	**Works**
Decision Tree (DT)	J48	Baneres et al. (2019), Hussain et al. (2018) , Jayaprakash et al. (2014)
	C4.5	Pierrakeas et al. (2020), AL-Malaise et al. (2014)
	/	Marbouti et al. (2016), Koprinska et al. (2015), Hussain et al. (2018), Yukselturk et al. (2014)
Neural Networks (NN)	Back Propagation (BP)	Pierrakeas et al. (2020),
	Feed Forward Neural Network (FFNN)	Yukselturk et al. (2014), Lykourentzou et al. (2009)
	MultiLayer Perceptron Network (MLP)	Marbouti et al. (2016), Huang and Fang (2013)
	Multivariate Adaptive Regression Splines (MARS)	Howard et al. (2018)
	Multivariate Adaptive Regression Splines (RBFN)	Huang and Fang (2013)
Support Vector Machines (SVM)	Radial Basis Function (RBF)	Mubarak et al. (2020)
	Sequential Minimal Optimization (SMO)	Pierrakeas et al. (2020),
	/	Baneres et al. (2019), Howard et al. (2018), Marbouti et al. (2016), Lykourentzou et al. (2009), Sandoval et al. (2018), Huang and Fang (2013), Jayaprakash et al. (2014)
K-Nearest Neighbors (KNN)	/	Baneres et al. (2019), Howard et al. (2018), Marbouti et al. (2016), Yukselturk et al. (2014), Wolff et al. (2014)

Table 2.3 – *Most used ML Techniques to predict learners' dropouts*

Learning algorithms or some personalized intelligent algorithms. Add to that the distributed nature of some treated problems, like the case of our research, requires the use of Distributed Artificial Intelligence (DAI) solutions like Multi-Agent Systems (MAS).

Unfortunately, we only found one work AL-Malaise et al. (2014) that combined Machine Learning with Multi-Agent System technology to predict the outcomes of the learners. That is why we used an agent-based solution combined with a custom-made algorithm to predict dropout learners in a short amount of time and with a small set of data.

Machine Learning Technique	**Works**
Naive Bayes Classifier (NBC)	Baneres et al. (2019), Marbouti et al. (2016), Yukselturk et al. (2014), Hussain et al. (2018), Jayaprakash et al. (2014)
Bayesian Profile Regression (BPR)	Sarra et al. (2019)
Bayesian Additive Regressive Trees (BART)	Howard et al. (2018)
Random Forest (RF)	Howard et al. (2018)
Logistic Regression (LR)	Marbouti et al. (2016), Waddington et al. (2016), Jayaprakash et al. (2014)
Classification And Regression Tree(CART)	Wolff et al. (2014), Hussain et al. (2018)
Bayesian Network (BN)	Wolff et al. (2014)
Gradient-Boosted Trees (GBT)	Hussain et al. (2018)
J-Repeated Incremental Pruning (JRIP)	Hussain et al. (2018)
Linear Regression (LR)	Sandoval et al. (2018)
Robust Linear Regression (RLR)	Sandoval et al. (2018)
Multiple Linear Regression (MLR)	Huang and Fang (2013)
Principal Components Regression (PCR)	Howard et al. (2018)
Probabilistic Ensemble (PE)	Lykourentzou et al. (2009)
Probabilistic Ensemble Simplified Fuzzy ArtMAP (PESFAM)	Lykourentzou et al. (2009)

Table 2.4 – *Other ML Techniques used to predict learners dropouts*

Chapter 3

Multi-Agent Systems In Human Learning Environments

Contents

3.1 Introduction 48
3.2 Distributed Artificial Intelligence 48
  3.2.1 *What is Artificial Intelligence?* 49
  3.2.2 *Distributed Artificial intelligence (DAI)* 50
3.3 What is an agent? 50
  3.3.1 *Definitions* 51
  3.3.2 *Intelligent/Rational Agents* 55
  3.3.3 *Classification of Agents* 56
  3.3.4 *Reinforcement Learning Agents* 61
  3.3.5 *Pedagogical Agents* 61
3.4 Multi-Agent Systems 64
  3.4.1 *Definition* 65
  3.4.2 *The Control Problem* 65
  3.4.3 *Mechanisms of Coordination* 66
  3.4.4 *Existing Multi-Agent System Platforms* 67
3.5 Intelligent Agents and Human Learning 67
  3.5.1 *Usage of Intelligent Agents in Education* 68
3.6 Conclusion 72

3.1 Introduction

Man has always dreamed of having machines capable of doing his work. However, despite the development of numerous mechanical systems that allowed him to lighten up his burden at the time, these machines were unable to function by themselves as they were deprived of intelligence. This situation remained unchanged until Alan Turing asked a daring question, "Can machines think?" (Turing, 1950). This question was the beginning of a new era that has lasted until today. We witnessed the rise of artificial intelligence, starting from the most straightforward applications like text translation to the most complicated surgical operations and even space conquest missions.

All these scientific breakthroughs would not be possible without the invention of computers. Nowadays, computers have made our lives easier. No one can imagine living without computers. Engineers, doctors, students, teachers, entrepreneurs, investors, and government agencies use them for daily tasks, entertainment, online earnings, and office work. Because computers have superior precision and quality, they spend less time and can accomplish tasks quickly, but performing the same operation by hand might take a long time. Computers have made it possible for industries and businesses to operate globally. Furthermore, they are present in every industry and aspect of life. Online education, hospitals, government administration, private and public businesses, and entertainment depend on computers.

However, computers are not particularly effective at anticipating, planning, and coding their actions. If a given software runs into a problem that its creator did not anticipate, it leads to unexpected results, varying from mere system crashes to human casualties. This fundamental reality underpins our connection with computers. Those familiar with computers know this, yet it is rarely acknowledged. It still surprises people who are unfamiliar with computers. We generally consider computers as dutiful, literal, unimaginative slaves. For many purposes (like payroll processing), it is okay. With an expanding number of applications, we need systems that can autonomously determine what to do to meet their design objectives. Agents are such computer systems. Intelligent agents, or autonomous agents, must function robustly in fast-changing, unpredictable, or open contexts where actions may fail (Wooldridge, 1999). Intelligent agents are widely used in industry, medicine, communication, and education.

This chapter presents an overview of intelligent agents and their use, especially in education, and how they are used to identify and predict learning difficulties. This chapter begins by providing broad definitions of Artificial Intelligence (AI) and Distributed Artificial Intelligence (DAI). It then passes to intelligent agents, their features, and classifications. It also introduces Multi-Agent Systems and their characteristics and their use and impact in education.

3.2 Distributed Artificial Intelligence

While "Artificial Intelligence" (AI) is one of today's most prominent subjects, it was first developed in 1950 and went through a hype cycle between 1956 and 1982. This section outlines some of the accomplishments made during the cycle's boom period and explains what led to the cycle's crash phase. It is essential not to miss the lessons learned from this hype cycle. Its achievements established the foundations for today's machine learning algorithms, but its flaws revealed the risks of overconfidence in promising research and development sectors.

3.2.1 What is Artificial Intelligence?

Artificial intelligence (AI) refers to the "*simulation of human intelligence by machines, primarily computer-based systems*". AI applications are now widely used in various industries, like education, healthcare, and manufacturing. Many technologies incorporate AI, like expert systems, natural language processing, speech recognition, and computer vision.

"Chinese Room Argument" (CRA) Searle (1980) is a mind experiment that has Searle imagine himself alone in a room, listening to a computer program to answer Chinese symbols slipped under the door. Although Searle understands nothing of Chinese, as he follows the program to process the symbols and numbers as a computer would, he replies with appropriate strings of Chinese characters under the door, leading him to mistakenly assume that there is a Chinese speaker in the room by those outside it (Cole, 2020). In his CRA, Searle (1980) divided AI into Weak and Strong.

- **Weak AI**, also known as "Narrow AI", is an AI system designed and trained to accomplish a particular job. Industrial robots and voice-activated assistants use weak AI, such as Apple's Siri, Microsoft's Cortana, Amazon's Alexa, or Google Assistant. It can answer a question or execute a programmed command but cannot function without human interaction.
- **Strong AI**, also known as Artificial General Intelligence (AGI), refers to programming that is capable of replicating the cognitive abilities of the human brain. When faced with an unfamiliar task, a Strong AI system can autonomously use fuzzy logic to find a solution. A self-driving car is an example of strong AI utilizing the combination of computer vision, image recognition, and deep learning to control a vehicle to stay in a given lane and avoid unexpected obstacles like pedestrians. In theory, a strong AI program should be able to pass both a Turing Test (Turing, 1950), and the Chinese room test Searle (1980).

As specified by Abioye et al. (2021), there are three types of AI, namely, Artificial Narrow Intelligence (ANI), Artificial General Intelligence (AGI), and Artificial Super Intelligence (ASI) (Figure 3.1).

- **ANI:** sometimes referred to as Weak AI (Baum et al., 2017). It refers to AI, where machines display intelligence in a specific domain, such as chess-playing, sales prediction, and movie recommendations (Franklin, 2007).
- **AGI:** This branch of study, sometimes known as Strong AI, is concerned with getting machines up to the same performance level as humans (Franklin, 2007).
- **ASI:** its mission is to create machines that outperform humans in several different disciplines (Baum et al., 2017). Many individuals are concerned about ASI because it predicts that machines may eventually exceed humans.

.

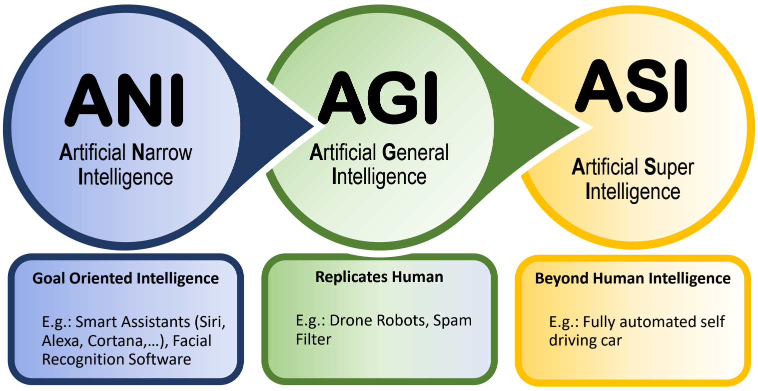


Figure 3.1 – *Types of AI (Abioye et al., 2021)*

3.2.2 Distributed Artificial intelligence (DAI)

Distributed Artificial Intelligence (DAI), also called Decentralized Artificial Intelligence, is a subfield of Artificial Intelligence dedicated to developing solutions for classic artificial intelligence problems that require analyzing large amounts of data by dividing and distributing the problem on independent and distributed processing nodes (agents). DAI is closely related to and a predecessor of the Multi-Agent Systems (MAS) field.

DAI operates in parallel; thus, it can exploit large-scale computational resources, and its properties allow it to solve problems that require enormous datasets processing. DAI systems consist of distributed, independent, and autonomous processing nodes (or agents). DAI is multidisciplinary by nature. Therefore, it took advantage of technological breakthroughs in other related domains such as artificial intelligence, distributed computing, social and organizational sciences, cognitive sciences, and philosophy (Ferber and Weiss, 1999; Moulin and Chaib-draa, 1996).

Weiss (1999) defines DAI as "the study, construction, and application of MAS, that is, systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks."

3.3 What is an agent?

An agent can be anything that can perceive its environment through sensors and act on it through effectors. A human agent contains sensors in the form of eyes, ears, and other organs and effectors in the form of hands, legs, mouths, and other bodily parts. The sensors are replaced by cameras and infrared range finders, while the effectors are replaced by different motors. Perceptions and behaviors of a software agent are encoded bit strings (Russell and Norvig, 2010; Lu and Wang, 2020).

Various fields have studied the topic of agents for decades. The agent is a general name for several entities used in several domains such as biological knowledge-based systems (called biological agents), autonomous robotics, computer software along its components integrated into operating systems or complicated computer systems, natural language processing, and other artificial intelligence applications. Agents were also used in philosophy and psychology Bakhta (2014).

3.3.1 Definitions

Wooldridge and Jennings (1995), defined an agent as a software or a hardware computer system that possesses these features:

- **Autonomy:** agents act independently of humans' intervention, and they demonstrate some control over their behaviors and internal state.
- **Social Ability:** agents can interact with other agents and humans using a communication language.
- **Reactivity:** agents can perceive their environment and react appropriately to any occurred changes.
- **Proactiveness:** in addition to simply reacting to their environment, agents are also able to initiate goal-directed behavior by taking the initiative.

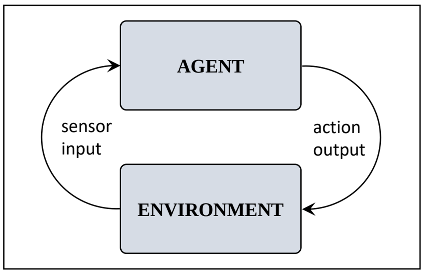


Figure 3.2 – *An Agent as specified By Wooldridge and Jennings (1995)*

Ferber (1997), defined an agent as a virtual, or a physical, entity that acts in an environment to achieve a set of objectives using available resources and skills; Using sensors, it could perceive the environment and make autonomous decisions to act on that environment or even on itself using actuators. Furthermore, it can communicate directly with other agents (Figure. 3.3).

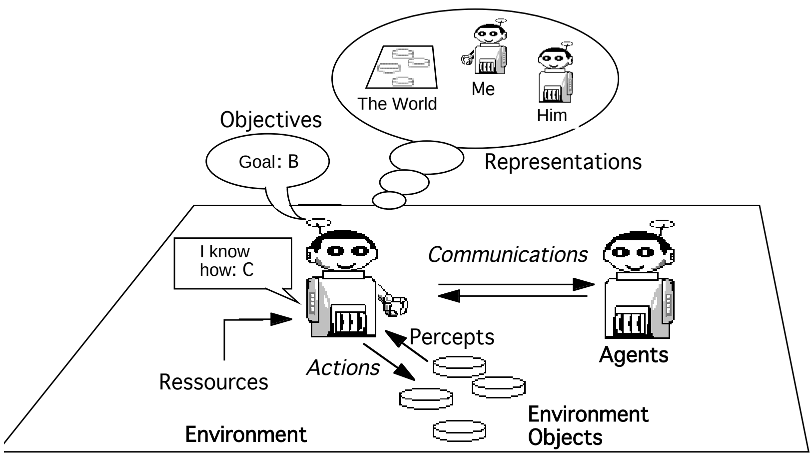
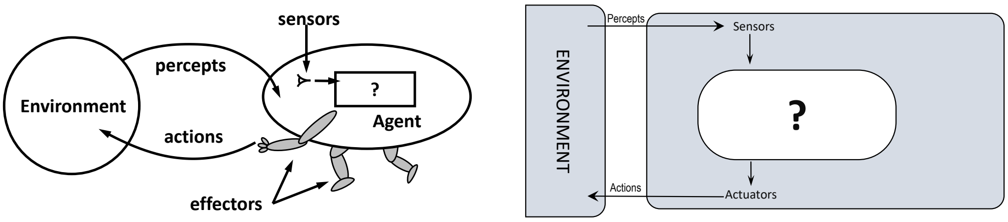


Figure 3.3 – *An Agent as specified By (Ferber, 1997)(Translated from French)*

Jennings and Wooldridge (1998) defined an agent as a computer system located in an environment that acts autonomously and flexibly to achieve the objectives for which it was designed. However, autonomy is a confusing concept and challenging to define correctly.

Russell and Norvig (2010), define An agent as anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. This simple idea is illustrated in (Figure. 3.4) .



(a) (b)

Figure 3.4 – *An Agent An Agent as specified By (a) (Russell and Norvig, 1995) and (b) (Russell and Norvig, 2010)*

We agree with Wooldridge (1999), who claimed that there is no commonly recognized definition of the term "agent", and there is a considerable dispute and disagreement around this topic. Essentially, while there is a broad agreement that autonomy is essential to the concept of agency, there is nothing else on which to agree. Part of the problem is that different domains appreciate distinctive features associated with agency differently. Thus, while the capacity of agents to learn from their experiences is critical in specific applications, it is not only irrelevant but even undesirable in others.

Nevertheless, combining common points of all these definitions, we could define an agent as:

> *A physical or a virtual entity that seeks to fulfill its design objectives and whose behavior is autonomous and evolves in an environment that it can perceive and in which it can act on it or on itself and can communicate with other agents or humans (Figure. 3.5).*

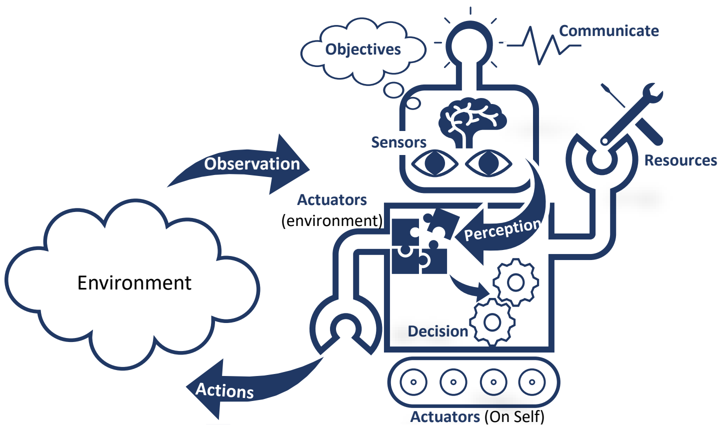


Figure 3.5 – *An Agent in its Environment*

However, as believed by Wooldridge (2013), all these definitions are only relevant for the “agent,” while the definition of an “intelligent agent” should be more detailed. Furthermore, in the opinion of the same authors, this broad and fuzzy definition does not specify what type of environment an agent occupies or gives a specific definition of the autonomy concept, which could be tricky (see Subsection 3.3.2).

3.3.1.1 Agents are Autonomous

Ferber (1997) argues that multi-agent systems are valid societies of beings that can move, plan, communicate, perceive, act and react and, in general, "live" and work in an environment in which they can even come into conflict with other agents. As declared by the same author, unlike most computer programs, multi-agent systems are not "thinkers" closed only to their reasoning and unaware of their environment.

The concept of autonomy underpins the whole agent paradigm. Therefore it is critical to clarify that concept. Jennings and Wooldridge (1998) considered it to indicate that a system should be able to behave without the interference of humans or other agents and that it should have control over its actions and state. (Russell and Norvig, 2010) stated that an agent lacks autonomy if its behaviors are entirely dependent on built-in knowledge and it does not need to pay attention to its perceptions. For example, suppose a clock builder was foresighted enough to predict that the clock owner would travel to Australia on a specific day. In that case, he could place a mechanism to automatically shift the clock’s hands by six hours at precisely the right moment. The clock’s behavior would undoubtedly be effective, but its intelligence belongs to its creator rather than the clock itself. The agent’s behavior can rely on its experience and information embedded into it when designed to function within a specific environment (Russell and Norvig, 2010).

As believed by Russell and Norvig (2010), a system is autonomous when it behaves in harmony with its own experiences. However, requiring total autonomy from the start would be too demanding: since the agent had little or no experience driving it to act randomly unless the designer provided some support. It would be logical to empower artificial intelligent agents with initial knowledge and learning ability. Autonomy does not only align with our instincts but also demonstrates suitable engineering methods. When an agent operates based on its built-in assumptions, it can only operate well when those assumptions are valid and lack flexibility. Given sufficient time to adapt, a genuinely autonomous intelligent agent should operate successfully in a wide variety of environments.

Typically, each agent has its available repertoire of actions. Nevertheless, not all actions are applicable in every case. For example, “lift desk” only applies when the desk’s weight is minimal enough for the agent to raise it. Similarly, the action “buy a car” may fail if insufficient funds are available. Actions thus have pre-conditions that specify the environment in which they can be used(Russell and Norvig, 2010).

3.3.1.2 Agents Has Environments

One of the agent’s main challenges is selecting which actions to perform to meet its design goals. Agents’ architectures are software architectures for embedded decision-making systems. Several environmental variables can influence the decision-making process’s complexity (Wooldridge, 1999).

3.3.1.3 Agents are Adaptive

An agent grows continually and independently in an ecosystem populated by other agents. As a result, an agent is distinguishable from its surroundings. The designer introduces the latter at the same time as the agent. The environment is a structural entity that can exist due to dynamic processes. An agent can thus be largely unknown or even hostile. Cohen and Levesque (1988) believe that an agent must evolve in its environment by design.

An agent's environmental attributes are connected to its representation of that same environment and of itself. An agent's relationship to its environment might be strong or weak. In the first scenario, the agent incorporates sensors allowing it to track the environment's transformations immediately. In mobile robotics, for example, we use sensors to study the robot's environment in three dimensions. In the second scenario, the agent learns about occurrences through non-controlled agents. The type of connection depends on the application domain. Some domains allow both forms of connections (Guessoum, 1996).

3.3.1.4 Agents and Objects

In programming, we define an object as a software entity that encapsulates a given state and is capable of executing actions called methods on that state. It is also able to communicate by passing messages. In addition to the state, agents encapsulate behavior compared to objects that have no control over the execution of their methods. For example, if object x executes a method m as asked by object y, then object x has no choice but to run that asked method m. In contrast, if agent y asks agent x to execute an action a, agent x assesses whether it serves its design purposes or not, then decides to run it or not.

In these terms, the object is not autonomous, as it has no control over its proper actions. Consequently, when an agent invokes another agent's method (action), this agent is requesting the execution of that agent's method. It is up to the recipient agent to decide whether to take the act or not on that request (Wooldridge, 1999).

Wooldridge (2013) summarized the differences between agents and objects in these three points:

- **Degree of autonomy**: agents emphasize a more profound notion of autonomy than objects as they decide for themselves whether or not to execute requested actions.
- **Degree of smartness**: capable of flexible (reactive, pro-active, social) behavior, while objects lack such behavior.
- **Degree of activeness**: by inheritance, a Multi-Agent System is multi-threaded in the sense that each agent is assumed to have at least one active control thread.

3.3.1.5 Examples of Agents

Wooldridge (1999) states that any control system is an agent. A thermostat is a simple yet overused example. Thermostats feature a temperature sensor. This sensor is implanted in the environment, like the room, and provides two signals, one for low temperature and one for ambient temperature. The thermostat has two actions to turn the heating ON or OFF. A heater would typically raise the room's temperature, although this is not assured if the room's door is open. The thermostat's (very simplistic) decision-making component follows the following rules:

$$ThermostatAgent : \begin{cases} \texttt{Temperature} < \texttt{Threshold (Cold)} & \rightarrow \text{Heating } \textbf{ON} \\ \texttt{Temperature} \geq \texttt{Threshold (O.K.)} & \rightarrow \text{Heating } \textbf{OFF} \end{cases}$$

More complicated environmental control systems have significantly richer decision structures like nuclear reactor control, space probes, or even auto-piloted airplanes.

As alleged by Jennings and Wooldridge (1998), most software daemons (such as background processes in the Operating System) may be thought of as agents; They keep an eye on a software environment and take actions to change it. The software daemons live in a software environment, unlike our thermostat agent in the previous example, which lived in a physical environment—the physical world. They get information about their surroundings by performing software operations, and the actions they take are software actions (changing an icon on the screen or executing a program). The decision-making process is similar to that of the thermostat.

To conclude, an agent is a computer system that can function autonomously in a given environment. To affect its environment, an agent often has a repertoire of accessible actions, which can be performed non-deterministically by physical sensors in the case of physical, real-world agents, or software sensors in the case of virtual agents(Wooldridge, 1999).

3.3.2 Intelligent/Rational Agents

An intelligent agent, as explained by Jennings and Wooldridge (1998), is a software entity that is capable of flexible, autonomous action to achieve its design goals. Authors define flexibility as the ability of a system to be:

– **responsive:** agents may be competent to perceive and react to changes in their environment in real time.

– **proactive:** agents should take part in strategic, goal-directed activities and take the initiative when required; they should do more than react to their environment;

– **social:** agents must be able to interact with other entities, agents, or humans when they judge it necessary to solve their proper problems but also assist others with their activities.

In that same direction, Russell and Norvig (2010) defined a "Rational Agent" as one that does the right thing. Undoubtedly, this is preferable to doing something wrong, but what does it imply? As a first approximation, the right action is the one that leads to the best agent's success, which leaves us with the dilemma of determining *how* and *when* to assess the agent's performance.

The same authors used the term "Performance Measure" to describe *how* an agent performs. The observer, outside the environment, has to set a benchmark for success in an environment and use it to evaluate each agent's performance. As an example, consider an agent cleaning the floor. The amount of dirt cleaned up in an eight-hour shift might be a performance indicator. A more advanced performance metric would also consider power usage and noise production. *When* to evaluate performance is also vital. To reward quick starters, even if they perform little or no work later on and penalize consistency, we may assess *how* much dirt the agent picked up in the first hour of the day. So we want to quantify performance throughout time, whether it is a day or a lifetime.

3.3.3 Classification of Agents

Same to the definition of the agent concept, there is no standard classification of the agents. Three scientists made the three famous classifications we found in the literature: Ferber (1997), Russell and Norvig (2010), and Wooldridge (2013)[1].

3.3.3.1 Ferber's Classification

Ferber (1997) groups agents into cognitive and reactive types, with teleonomic (goal-directed) and reflexive action (governed by perceptions).

1. Reactive Agents: The Perception/Action cycle governs the behavior of reactive agents (Figure 3.6). Their behavior is a finite state machine establishing a link between sensory input and an output action. As alleged by Drogoul (1993), reactive agents have no memory capacity. As a result, they possess no representation of their environment or other agents.

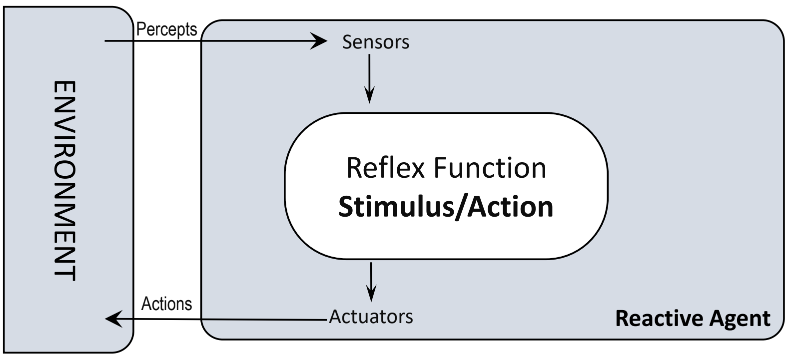


Figure 3.6 – *Reactive Agent Diagram (Ferber, 1997)*

2. Cognitive Agents: The human model is used to create cognitive agents. Each agent has a partially accurate representation of the environment and the other agents. They act in a Perception/Decision/Action cycle(figure 3.7). Each agent may reason based on its own goals and information about the other agents. Furthermore, the agents may communicate in a human-like discussion manner. Cognitive agents, in contrast to

[1] Wooldridge, M. wrote this chapter book included in G. Weiss's edited book in 2013. However, somehow it is sometimes referenced as "G. Weiss (2013)."

reactive agents, can carry out sophisticated processes and solve issues by themselves (Drogoul, 1993).

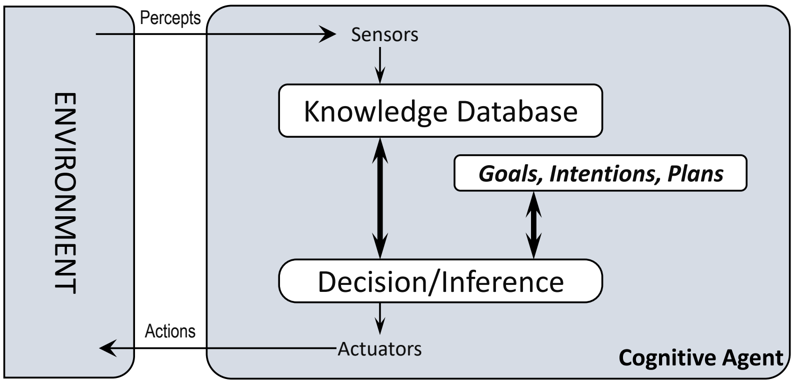


Figure 3.7 – *Cognitive Agent Diagram (Ferber, 1997)*

3. Hybrid Agents: Some researchers have focused on the creation of hybrid agent models that mix reactive and cognitive capacities to answer the challenge of adjusting cognitive agents' behavior to the evolution of the environment in real-time (Guessoum, 1996). This allows the benefits of each of the two prior models to be balanced against their shortcomings. A hybrid agent's behaviors are influenced by its goals rather than external inputs.

3.3.3.2 Russell's Classification

Russell and Norvig (2010) divided agents into five categories.

1. Simple-reflex Agents.
2. Simple-reflex Agents With Internal State.
3. Goal-directed Agents.
4. Utility-based Agents (Usefullness).
5. Learning Agents

1. Simple-Reflex Agents: This type of agent is based on condition-action rules, Also called situation-action rules, productions, or if-then rules. For example, if the automobile in front of you brakes and its brake lights are on, the driver should be aware of this and commence braking. In other words, some processing is done on the visual input for generating the condition "The automobile in front is braking," which then triggers some previously established relationship in the agent program to the action "initiate braking." The condition-action rule could be stated as:

if *car-in-front-is-braking*, **then** *initiate-braking*.

Figure 3.8 depicts the schematic layout of a Basic Reflex Agent, demonstrating how the condition-action principles allow the agent to make the connection from percept to action.

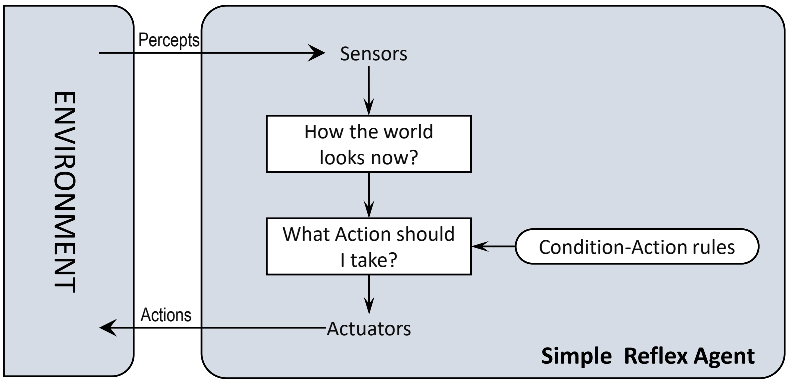


Figure 3.8 – *Simple-reflex Agent Diagram* (Russell and Norvig, 2010)

2. Simple-Reflex Agents With Internal State: The simple reflex agent described previously would only operate if the proper decision can be made based on the present percept.

Consider the following situation: The driver checks the rear-view mirror from time to time to see where other vehicles are. The vehicles in the next lane are unseen while the driver is not looking in the mirror; nonetheless, to decide on a lane-change move, the driver must know whether or not they are present.

This example illustrates a difficulty that emerges when the sensors do not offer access to the entire state of the world. In such instances, the agent may need to save some internal state information to differentiate between world states that produce the same perceptual input but differ dramatically. Updating this internal state information over time necessitates the inclusion of two types of knowledge in the agent program.

- First, we need to know how the world evolves in the agent's absence.
- Second, we need to know how the agent's activities influence the rest of the world.

The structure of the reflex agent is depicted in Figure 3.9, which shows how the current percept is coupled with the previous internal information to provide an updated description of the present state.

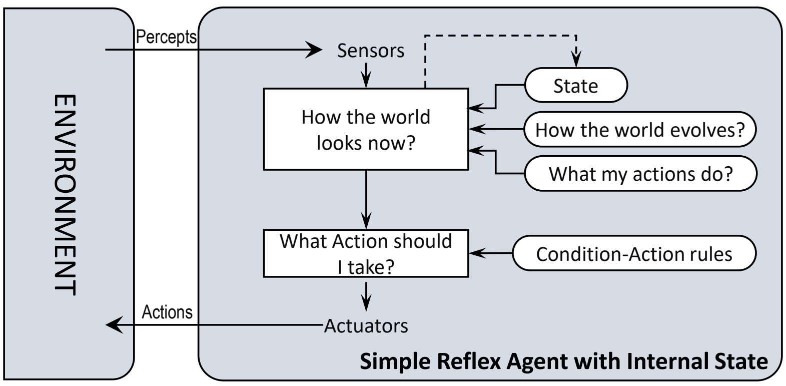


Figure 3.9 – *Simple-reflex Agent with Internal State Diagram* (Russell and Norvig, 2010)

3. Goal-Directed Agents: Knowing the current status of the environment is not always sufficient to make decisions. For example, the taxi driver can turn left, right, or continue straight at a road intersection. The best option is determined as required by the passenger's destination. In other words, in addition to a present state description, the agent needs "goal" information, which defines desired scenarios, such as being at the passenger's destination. The agent program can combine that information with knowledge about the outcomes of various actions to choose actions that fulfill its design objectives, which is the same information used to update the internal state in the reflex agent.

Unlike the condition-action principles stated previously, this decision-making implies thinking about the future to answer both questions "What would happen if I do this?" and "would that make me happy?" As a result, the designer has to precalculate the proper action for numerous circumstances.

The goal-based agent is less efficient but far more adaptable. If it starts to rain, the agent may update its understanding of how well its brakes work, changing all necessary actions to match the current conditions. For the reflex agent, we would have to modify several condition-action rules. Of course, the goal-based agent is more adaptable to diverse destinations. By just changing the goal-based agent's destination, we may change its behavior. Its rules for turning and straightening out only work for one location; they must all be replaced to travel somewhere else. Its structure is shown in Figure 3.10.

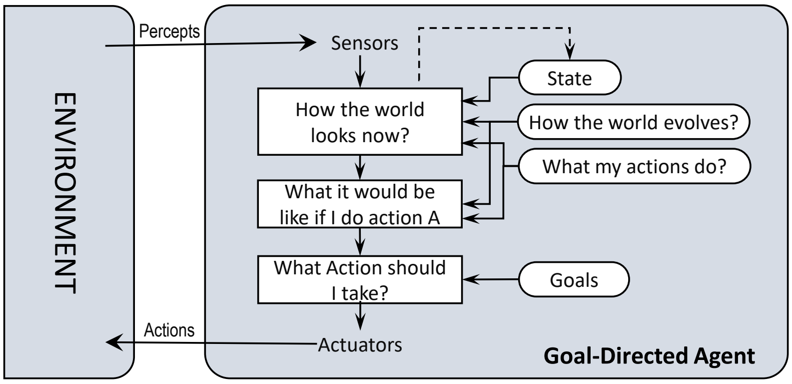


Figure 3.10 – *Goal Directed Agent with explicit goals* (Russell and Norvig, 2010)

4. Utility-based Agents (Usefulness): Goals alone do not produce excellent behavior. Following the previous example, several action sequences would bring the taxi to its destination, but some are faster, safer, more dependable, or cheaper. A more generic performance metric should allow comparing alternative environment states, or sequences of states, based on how pleased they would make the agent if attained. Because "happy" is not a scientific concept, it is commonly said that a world condition has better utility for the agent.

The utility is a function that translates a condition to a number that describes the related degree of joy. A comprehensive utility function formulation facilitates reasonable judgments in two types of goal-confusion instances. First, the utility function determines the right action when desired goals, such as speed and safety, can only be partially attained. Second, when an agent has several objectives, none of which can be attained with certainty, the utility function allows the agent to measure the possibility of achievement against the goals' relevance.

An agent with an explicit utility function may make decisions but must compare the utilities obtained by various behaviors. The agent can choose immediately an action if it meets the goal. In certain circumstances, the utility function may be transformed into a set of objectives, allowing a goal-based agent to make the same decisions as a utility-based agent. Figure. 3.11 depicts the utility-based agent organization.

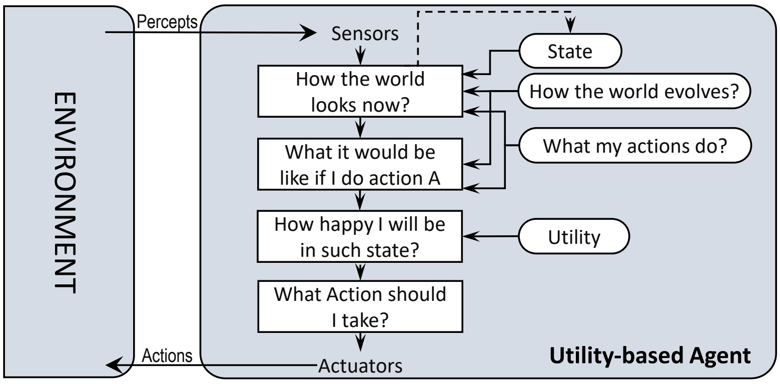


Figure 3.11 – *Complete Utility-Based Agent (Russell and Norvig, 2010)*

5. Learning Agent[2]: It utilizes four abstract concepts, which are:

- **Learning Element:** Helps make improvements.
- **Performance Element:** Selects external actions.
- **Critic:** Provides feedback that the learning element uses on how the agent is doing and how performance elements must be modified.
- **Problem Generator:** Suggests actions that lead to new and informative experiences.

For example, commercial virtual assistants like Siri[3] and Alexa[4] to predictive searches, all use learning agents to adapt and learn about the user, make accurate suggestions and recommendations, and deliver relevant ads.

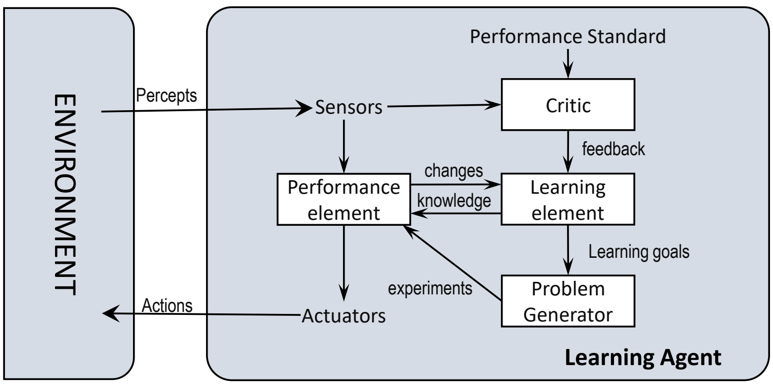


Figure 3.12 – *Learning Agent (Russell and Norvig, 2010)*

[2] To not be confused with Reinforcement Learning Agent which is related to ML reinforcement Learning (see Subsec 3.3.4)

[3] Siri *https://www.apple.com/siri/* (Accessed: 26, May 2022)

[4] Alexa *https://www.alexa.com/* (Accessed: 26, May 2022)

3.3.3.3 Wooldridge's Classification (also known as Weiss's Classification)

In the book edited by G. Weiss, Wooldridge (2013) considered four classes of agents:

- **Logic-Based agents:** It decides what action to take based on logical deduction.
- **Reactive agents:** It uses decision-making mechanisms implemented in the form of direct mapping from situation to action.
- **Belief-Desire-Intention (BDI) agents:** Its decision-making is based on the manipulation of data structures representing the agent's beliefs, desires, and intentions; and ultimately.
- **Layered architectures:** Decisions are made through many software layers, each of which is more or less overtly reasoning about the environment at varying levels of abstraction.

3.3.4 Reinforcement Learning Agents

The Reinforcement Learning Agent (RL Agent) is a type of Machine Learning in which the agent trains by interacting with its environment via trial and error. It has learning capabilities and can learn from its past experiences. It does not follow a similar agent program like the other agents, as it starts from the basic knowledge and then acts and adapts through learning (Hook et al., 2021; Subagdja et al., 2009). It has attracted considerable interest from the research community due to its potential to learn decision-making through the abstraction of experiences rapidly (Hook et al., 2021).

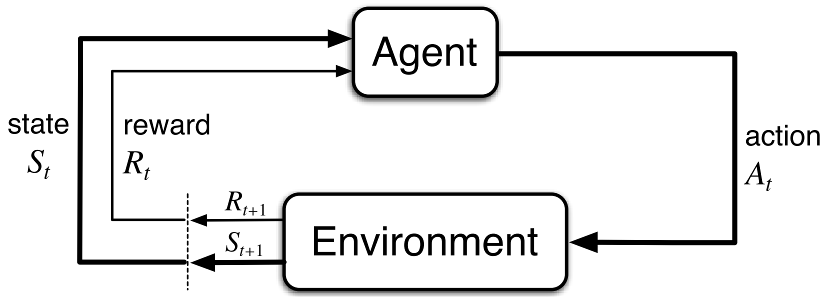


Figure 3.13 – *A Reinforcement Learning Agent (Sutton and Barto, 2012)*

Reinforcement learning (RL) is a branch of ML that studies how intelligent agents should act in a given environment to maximize the concept of cumulative reward. Reinforcement learning, along with supervised and unsupervised learning, is one of the three basic machine learning paradigms.

3.3.5 Pedagogical Agents

As stated by Clarebout and Heidig (2012), "Pedagogical Agents" are animated characters that guide users through multimedia learning environments on a computer screen. In educational software, they are referred to as "learning partners" or "virtual tutors" following their work. As stated by the same authors, the most often used theoretical framework for studying pedagogical agents is social agency theory (Mayer, 2005). The voice and image of the pedagogical agent offer social signals that drive social responses pushing them to process materials deeply. Therefore, social agency theory still refers to a facilitative impact due to the pedagogical agent's simple presence. Domagk (2010) argues that the valence of other social cues, such as sympathy for the character, is crucial.

In his book, Woolf (2008), defines "Animated Pedagogical Agents" as intelligent computer characters that help learners learn how to move around in an environment and look natural. Unlike simple cartoon characters, pedagogical agents have goals and decide how to attain them using AI to rationalize their acts. The author argues that non-teaching AI agents can operate autonomously in dynamic and uncertain contexts while pedagogical agents captivate learners with their unique personalities, life stories, and skills. They can be built to be "cool" teachers and develop, learn, and change as often as necessary to keep learners' attention and motivation. Human-like mood and behavior algorithms may help them find the best or most recent stuff. The visual representations, faces, and bodies, of physically embodied creatures are used to communicate and sense environmental stimuli like keyboard input, mouse position, or mouse clicks.

3.3.5.1 Agent Notion Disambiguation

Erickson (1997) stated that the term "agent" has two meanings that are frequently confused

- **a. An Agent-Metaphor,** which represents a character on the screen; and
- **b. An entity with Adaptive Functionalities,** which is a form of AI implemented in software applications but are not always visible on the screen (e.g., intelligence, adaptively, responsiveness).

Erickson (1997) argues that considering intelligent entities on the screen makes both meanings relevant. However, the distinction is significant when addressing educational agents. The phrase "pedagogical agent" refers to the agent metaphor, but not always to adaptive functions. Instructional agents frequently communicate using predefined texts with no artificial intelligence. This might be considered a limitation or an opportunity to see if presenting a character on the screen without their ability to respond provides additional learning benefits. Furthermore, "animated pedagogical agents" are often used interchangeably with "pedagogical agents" (Moreno, 2005). However, it ignores the fact that educational agents might also be static images or recordings of human tutors (Clarebout and Heidig, 2012).

3.3.5.2 Impact of Pedagogical Agents use

As determined by VanLehn (2011), studies on pedagogical agents fall into one of three categories: (1) studies on the presence of the agent, (2) studies on appearance or observable features, and (3) studies on actions or pedagogical practices. It is difficult to generalize pedagogical agents' effectiveness because their design, role, interactivity, and use vary widely. Studies on pedagogical practices and policies are less common because they do not require an embodied agent.

Many studies have been conducted to analyze the effects of pedagogical agents on learners' cognitive and affective outcomes. We grouped these studies into:

a. Cognitive Outcomes,

b. Affective Outcomes,

c. Self-Perception and Feelings Toward Learning, And

d. Motivation, Interest, and Self-Efficacy.

a. Cognitive Outcomes

A major concern in evaluating pedagogical agents' effectiveness based on their role in facilitating learning is that ITSs have historically been already successful without any extra components, which suggests there may be very little room left for improvement in terms of cognitive gains (Anderson et al., 1995; Vanlehn et al., 2005). Incorporating additional interactive components in a virtual learning environment may only increase cognitive load because it competes for the learner's attention. Adding animated agents has associated costs. So, what is the payoff? Do they enhance learning?

A two-decades-old review found no clear evidence for using pedagogical agents (Gulz, 2004), while at least two more reviews are ambiguous Heidig and Clarebout (2011); Veletsianos and Russell (2014). These papers illustrate various discrepancies between agent impact hypotheses and experimental data. Schroeder et al. (2013) argues that using pedagogical agents promotes learning owing to social interaction between a learner and an agent. More recently developed technologies like motion capture and animation may alter the nature and conclusions of studies covered in these assessments. The general topic of whether or not to include an agency has been less effective in recent evaluations. Not only does the technology crucial, but also how, when, and under what conditions it is employed. On the other hand, a meta-analysis by Schroeder et al. (2013) indicated that agents that use onscreen text rather than narration have bigger impact sizes. For some learners and in specific contexts, deploying pedagogical agents seems to yield small improvements; however, the learners and situations for which they are used remain to be determined (Gulz, 2004; Schroeder and Adesope, 2014).

b. Affective Outcomes

Pedagogical agents have been studied for their effects on learner perceptions, beliefs, and emotions. Most researchers agree that embodiment as a tutor, or co-learner, could enhance learning. Enabling a machine to replicate emotions like empathy and passion, researchers hope, will improve learning conditions.

Social agency theory (Atkinson et al., 2005; Krämer and Bente, 2010) is one of the primary reasons academics feel educational agents should impact the affective outcomes suggesting that social cues from an agent induce a kind of obligation to respond and interact. If pedagogical agents can engage learners socially and encourage them to pursue meaningful goals, then motivation should increase. The fundamental work on pedagogical agents seemed to support this hypothesis. Lester et al. (1997) hypothesized and examined the persona effect, stating that "the appearance of a lifelike figure in an interactive environment" can improve students' views of their own learning experiences. Moreno and Mayer (2000) offered further evidence for the concept, but these early investigations were challenged for lacking adequate controls (Clarebout et al., 2002).

c. Self-perceptions and Feelings toward Learning

When considering long-term goals like job choices and lifelong learning, the way learners perceive themselves or their learning content either positively or negatively is relevant. Schroeder and Adesope (2014) found that learners prefer having an agent over not having one (Baylor, 2009; Moreno and Flowerday, 2006). Earlier studies affirm that pedagogical agents increased good attitudes about the learning environment and increased learning effectiveness (Moundridou and Virvou, 2002; Atkinson, 2002).

The politeness effect is defined as the impact of face-saving strategies on learning. It has improved learning outcomes and learner self-perceptions in multiple settings and audiences (McLaren et al., 2011; Wang et al., 2008). There is also evidence that educational agents can help students learn more independently (Azevedo and Hadwin, 2005; Graesser and McNamara, 2010). However, pedagogical agents appear to be able to influence student feelings and views positively.

d. Motivation, Interest, and Self-efficacy

The learners' motivation to acquire and retain information is a significant question for educational technology producers. For this reason, research about pedagogical agents typically proposes them as a potentially transformative technology in terms of building engagement. Unfortunately, there is little empirical research on specific engagement assumptions (Domagk, 2010). A much broader topic is how an agent's presence affects learner motivation. Like our findings on cognitive outcomes, our findings on affective outcomes are mixed. Pedagogical agents can improve learners' computer-based learning experiences, or at least some of them.

Self-efficacy, or confidence in one's ability to fulfill activities or goals Bandura et al. (1999), is closely linked to motivation. The intrinsic rewards of knowledge are toyed with as learners pursue and master learning tasks. Like human coaches and teachers, the presence of an agent boosts self-reported belief in topic utility and self-efficacy in topic achievement (Rosenberg-Kima et al., 2007). Also, social affordances of agents such as speech and nonverbal actions may boost good emotion and learning transfer (Krämer and Bente, 2010). Agents have lately been connected to learning-friendly emotional states. An agent's passion and "energy" have increased learners' self-efficacy in informal learning contexts (Lane et al., 2013).

3.4 Multi-Agent Systems

As the name implies, Multi-Agent Systems (MAS) comprise several agents. In the preceding subsection, we defined the agent as a single entity that can operate autonomously to solve an issue. However, most issues demand more than that to be solved in the actual world. Applications capable of tackling complicated issues require interaction between numerous entities to attain that aim. In the "Agent programming" domain, we call them Multi-Agent Systems (MAS).

3.4.1 Definition

MAS is a collection of agents concurrently perceiving and acting in an environment. The theory is that an agent can be more effective when working with other agents by concentrating on tasks within its competencies while delegating other tasks beyond its abilities. It could use its skills to communicate, collaborate, cooperate, coordinate, and negotiate to solve problems that are beyond the individual capabilities or knowledge (Virginia and Julian, 2013).

Ferber (1997) defines MAS as communities of grouped agents that interact together to collaborate and coordinate their behavior to attain a common goal. They are made up of the following components:

- **An environment (E):** It is the environment where the agents are placed. The environment is only partially visible to each agent. In most cases, this perspective corresponds to the zone surrounding the agent whose range is equal to its sensing capability.
- **A collection of objects (O):** That are located in the environment E. They are passive, meaning that the agents may perceive, produce, destroy, and modify them.
- **A set of agents (A):** Which are specific objects ($A \subseteq O$) that represent the system's active entities;
- **A set of relations (R):** That connect the objects, and thus the agents.
- **A set of operations (Op):** That allow the agents of A to perceive, produce, consume, transform, and manipulate the objects of (O)
- **A set of Operators:** They are in charge of representing the application of these operations and the world's reaction to the attempted modification.

In her work, Sycara (1998) describes MAS as systems characterized with:

1. Limited viewpoint for each agent as it has incomplete information or skills to solve the problem individually;
2. No global control of the system ;
3. Decentralized data and
4. Asynchronous computation.

3.4.2 The Control Problem

The control, or the decision, of which entity to activate at each phase of the problem-solving process, is a crucial issue in multi-agent systems, either centralized or decentralized. The control is centralized if a higher-level entity has a worldwide perspective of the system's operations. In contrast, the control is decentralized if no single entity has a global picture of the system's operations. In that case, each agent in the system is in charge of its proper activities and contacts with other agents (Guessoum, 1996).

3.4.3 Mechanisms of Coordination

There are numerous techniques for coordinating the behaviors of different agents in the multi-agent literature. Durfee et al. (1987) offered three strategies for improving the coherence of a network of entities with distributed control. The final two techniques are utilized to boost coordination efficiency. To these three mechanisms, Gasser and Briot (1992) adds a fourth:

1. planning,
2. organization, and
3. meta-information sharing.
4. explicit analysis and synchronization

a. Planning: There are two forms of planning in multi-agent systems: centralized and distributed. A global picture of the plan is assumed in centralized planning when a central agent follows and analyzes all the other agents' behaviors so it can resolve their disagreements by defining a strategy that applies to all agents. The plan establishes the sequence of events that the other agents must follow. Such an agent does not exist in distributed planning where each agent develops its proper strategy. To detect and avoid conflicts, agents communicate their preliminary plans to each other.

The centralized method avoids conflicts and easily handles inconsistencies, but it is not well-suited for application domains with distributed agents. The distributed method is preferable in this scenario, but there is still much work to avoid disputes, manage discrepancies, and create an action plan.

3.4.3.1 Organization

A group of agents working together to complete one or more tasks is referred to as an organization. The organization is defined by [Gasser et al. 1989] as a set of settled or unresolved questions concerning beliefs and behaviors through which actors see other agents.

3.4.3.2 Meta-Information Sharing

This method is used to increase coordination efficiency. Control-level information on the entity's state and priority is known as meta-information. This sort of data can impact the control decisions of other agents who receive it. This information, like control knowledge, serves as a criterion for making judgments.

3.4.4 Existing Multi-Agent System Platforms

There are several MAS Platforms in use. Each platform has its specific tools, development methodologies, and applications. Most of the platforms offered to use Java programming language, allowing cross-platform development of multi-agent applications. Most multi-agent platforms are open-source and free.

Table 3.1 lists the most popular multi-agent platforms and their features.

#	Platform	Development tools	Main characteristics
1.	JADE (2000)	- Agent Management System - Director Facilitator - Agent Communication Channel	- FIPA-ACL Language
2.	MadKIT (1997)	- Agent/Group/Role model - Organisation-centered multi-agent System	- AGR organizational Model
3.	JACK (1997)	- Belief-Desire-Intention Model - JACK Development Environment (JDE) - JACK Object Modeller (JACOB) - JACK Plan Language (JPL) - Agent Run-time	- BDI model -Agent-oriented extension for Java
4.	Agent Builder (2004)	- Project Manager - Ontology Manager - Agency Manager - Agent Manager - Protocol Manager	- Ontology based - Knowledge Management Language (KML) - Knowledge Query and Manipulation Language (KQML)
5.	Cougaar (1997)	- Blackboard - HTTP servlet engine - Knowledge representation system	- large-scale applications - HTTP protocol
6.	Zeus (2000)	- Three layers of agents	- Role Modeling
7.	MASON (2003)	- 2d and 3d Libraries	- Low resources - 2D and 3D visualization.

Table 3.1 – *Most popular Multi-Agent Plateforms*

A description of these platforms, alongside a list of less-known platforms, is provided in Appendix. A.

3.5 Intelligent Agents and Human Learning

Several ITSs and e-learning systems have historically been established to apply artificial intelligence and cognitive science in education (Dolenc and Aberšek, 2015). AI technologies, including Machine Learning (ML), Expert Systems (ES), and Artificial Neural Networks (ANN), have been employed in these systems to assist instructors by detecting learners' learning styles, rating learners' learning performance, answering learners' questions, and providing simulated learning environments (Terzieva et al., 2021).

Intelligent agents can significantly assist individuals in these situations by reducing information overload and improving their performance. Moreover, they help in online shopping, for example, by saving time when searching for products and improving the quality of their decision-making (Sànchez-Marrè, 2022). Also, the applicability of intelligent agents has been historically investigated in numerous fields like e-commerce (Liang et al., 2019); knowledge extraction in distributed environments(Mosavi et al., 2018); and other knowledge-related tasks (Zhang et al., 2011a).

In e-learning systems, intelligent agents have proven very effective. While some online instructors still handle their online learners as if they were in a classroom, and many of them are too busy answering learners' inquiries to think about any new teaching pedagogy. Intelligent agents can assist e-learning learners in improving their learning outcomes while increasing their satisfaction with their online education experiences.

Liu et al. (2005) claim that online teachers have four primary roles in remote education:

– **Pedagogical Role:** Encouraging learners to share and build on their knowledge through discussion, creating different learning experiences, giving feedback, and referring to outside resources or experts in the field;

– **Social Role:** Providing a friendly atmosphere and a sense of community to enhance the cognitive learning processes of learners;

– **Managerial Role:** Managing assignments, online discussion forums, and overall course structure; and

– **Technical Role:** Guiding learners to resources for technical support, answering technical questions, diagnosing and explaining problems, and giving learners enough time to learn new programs.

Over the years, intelligent agents have taken over some teachers and educators as online facilitators for dull and repetitive tasks like checking computer codes, answering simple queries, reminding learners to turn in projects, and even grading essays. Also, intelligent technologies have assisted instructors in determining the best teaching strategies for each class or individual student based on student input (Elaachak et al., 2015).

3.5.1 Usage of Intelligent Agents in Education

Found works related to the use of intelligent agents in the educational field are dispatched into two major categories:

- Intelligent agents to support Learners.
- Intelligent agents to support Teachers.

3.5.1.1 Intelligent Agents to Support Learners

Using intelligent agents to help learners study in more dynamic and mobile contexts than their instructors may be crucial to their decision to seek online education rather than traditional classroom instruction. Especially for online learners who are burdened with several repetitive but crucial tasks like tracking deadlines, progress, and communication with group members, intelligent agents can perform these functions autonomously on their behalf (Muangprathub et al., 2020; Fellmann et al., 2020; Terzieva et al., 2021; Hensley et al., 2018; Rivera et al., 2018; Wan and Niu, 2018; Holstein et al., 2018).

a. Personalized and Adaptive Assistance: Rogers (1995) stated that humans have varying levels of learning effectiveness and efficacy. They can be divided into five categories based on their behaviors and skills toward innovation and originality, and their learning curve, in increasing order in concordance with their resistance to change: Innovators, Early Adopters, Early Majority, Late Majority, and Laggards (García-Avilés, 2020; Sääksjärvi and Hellén, 2019; Dale et al., 2021; Mallinson, 2021; Weil, 2018).

Take any computer-based learning environment, where many learners do not know how to select the appropriate tools to support their learning process; learners may find it difficult to make such a decision when there are numerous tools available. Intelligent agents can recommend the best learning techniques to learners based on their learning styles (Tsai et al., 2022; Muangprathub et al., 2020; Belkhadir et al., 2019).

To assist with online education, the instructor can offer learners an online test to help them identify their stronger primary learning styles. Intelligent agents can recommend the right learning style to each student based on his or her learning preference to improve overall learning effectiveness. For Visual learners, for example, the agent can recommend specific materials to help build the concepts about the prescribed learning content (Wan and Niu, 2018; Bodily and Verbert, 2017; Rivera et al., 2018; Çano and Morisio, 2017).

b. Effective Learning Time Management: Online learning success depends on effective time management. There is a clear link between their time management abilities and final grades. Learners with poor time management skills frequently miss critical deadlines for assignments and projects or fail to reply to requests from group members, resulting in tension within the study group (Razali et al., 2018; Hensley et al., 2018; Carlyon and Opperman, 2020).

An intelligent agent can assist online learners in better managing their time by alerting them to assignment due dates and reminding them of appointments. For example, if learners are very busy on Mondays, the agent may suggest that they do small assignments such as responding to others' posts; if learners have free time on Fridays, the agent may indicate that they do assignments that require long periods of concentration, such as writing the final project (Saleh et al., 2021; Fellmann et al., 2020).

c. Smart Mentoring for Effective Learning: The agent can assist learners in prioritizing tasks and creating effective work plans based on the amount of time they will likely spend on them, their level of complexity, their proximity to the due date, and their task-solving talents. For online group work to be successful in online education, learners must maintain constant communication with group members. Poor group work results when a student uploads an initial project proposal for feedback from group members but does not receive any on time. Intelligent agents can help avert this predicament by sending short messages to other group members' cell phones to remind them that their feedback is needed (Schwabe, 2021; Pammer-Schindler, 2020).

d. Intelligent-Group Formation: Intelligent agents can make it easier to find partners based on their shared interests, enabling knowledge sharing. For example, intelligent agents can gather learners with similar study preferences to facilitate cooperation and collaboration within one team project. Furthermore, team performance is an essential key element in improving learners' learning efficacy (Zhang et al., 2011b).

Online teachers frequently assign study sessions and project subjects without considering learners' specific study interests, leading to issues and a loss of motivation to study. Using AI technology for successful group learning, a computer-supported heterogeneous grouping system can help create small study groups based on individuals' personality attributes using AI technology for successful group learning (Maqtary et al., 2019).

Furthermore, in a networked society, a realistic simulation of distributed intelligence and multiagent systems can keep track of what each person understands, promoting prospects for productive collaboration and learning (Terzieva et al., 2021). As a result, learners can benefit from each other's expertise to improve their learning efficiency. When online learners have specific concerns and seek to receive solutions from the class, intelligent agents can locate qualified learners who might be ready to assist. Learners who require assistance will no longer have to look for it among their classmates.

3.5.1.2 Intelligent Agents to Support Teachers

Intelligent agents can free online educators from numerous tedious and time-consuming tasks, allowing them to concentrate on more creative and instructional work (Casamayor et al., 2009; Mikic Fonte et al., 2012). They can, for example, identify or predict learning styles or difficulties faced by learners, remind teachers to reply to unanswered matters, or even answer direct FAQs.

a. Extraction of Learning Styles and Thinking Styles: Individual qualities have a significant impact on human learning and academic achievement (Abood et al., 2020). The main five personality qualities in psychology, such as neuroticism, extraversion, openness to experience, agreeableness, and conscientiousness, for example, have predictable patterns of interactions with thinking styles (Zhang, 2007; Abe, 2020), and they affect academic success in school more or less (Furnham et al., 2005; Bratko et al., 2006; Van Bragt et al., 2011; Caspi et al., 2006). The learning styles of students are extremely important in their remote learning performance (Shahabadi and Uplane, 2015; Khamparia and Pandey, 2020). Each classroom has its own unique dynamic, which includes personalities, motivation levels, talents, and other factors (Truong, 2016; Chung and Ackerman, 2015). Balanced challenges and appropriate support for the readiness and ability of the learner are key ideas for an excellent educational approach. (Jaggars and Xu, 2016; Blake, 2009). The more a teacher understands about each student, the more he or she may tailor instructional tactics to the needs and goals of each student (Tkalčič and Chen, 2022; Tkalcic and Chen, 2015).

Because there is no face-to-face contact between instructors and learners in online distance education, instructors cannot utilize traditional approaches such as observation and intuition to detect individuals' personalities and learning styles (Shahabadi and Uplane, 2015; Truong, 2016; Chung and Ackerman, 2015; Bratko et al., 2006; Zhang, 2007). Intelligent agents can automatically assess student behavior by analyzing data such as learners' login time, courseware browsing sequence, how they perform in their evaluations and assignments, how quickly they return them, and the number of questions posed to each forum to determine student behavior; this will assist instructors in developing individualized teaching tactics (Latham et al., 2010; Pham and Adina, 2013; Sun et al., 2007).

b. Intelligent Learners' Feedback Management: Online learners expect quick replies to their questions, comments, and assignments, as well as teachers who are available at all times (Zeiser et al., 2018). Instructors frequently build web forums for the learners to post their specific questions and inquiries regarding existing courses. Instructors who respond quickly improve their teaching abilities and learners' satisfaction. On the other hand, some teachers overlook the message board for one or two days, causing learners to wait longer for answers to their questions. Intelligent agents can help continuously monitor learners' forum postings and alert instructors for new questions that require their response. Instructors can answer simple queries immediately or later, but they must tell learners that a response will be provided soon. Because of the agent support, the teachers are highly accessible to the learners (Holstein et al., 2019; Colace et al., 2018; Holstein et al., 2018).

c. Management of Frequently Asked Questions (FAQs): FAQ agents can answer learners' questions asked in natural language by matching question-answer pairs. When courses are delivered repeatedly, numerous questions are asked over again. Intelligent agents can organize these questions and related issues into a FAQ database. Many questions can be automatically fetched and answered by the intelligent FAQ agent. This can save time for instructors by eliminating the need to repeat tasks. Instructors will also have more time to address new difficulties and create new course strategies. By looking at which questions are answered incorrectly by most learners on their first attempt, the intelligent agent may identify the problematic and perplexing questions. This informs educators about which issues may require additional explanation for learners to absorb the material properly. This data can then be saved in the FAQ link for future reference (Khin and Soe, 2020; Reyes et al., 2019).

d. Identifying Struggling and At-Risk Learners: To learn more effectively and, if possible, catch up to their peers in terms of academic achievement and social development, learners require close attention and targeted care and assistance. Nevertheless, to offer such aid, instructors need to know their learners' needs in terms of help and support, as each one of them needs a different type and amount of assistance depending on their level of struggle. To do that, it is required to identify struggling learners in the first place to know how and when to target each one with the right kind and amount of assistance. Many strategies can assist teachers in identifying learners who are having learning difficulties. Instructors can offer a certain amount of assignments such as quiz questions to learners to review their understanding, allowing them to identify learners who score poorly on these questions. Students' performance can also be assessed using AI technics based on their traces. Intelligent agent systems can be equipped with ML algorithms or EDM techniques such as text-mining, for example, to create individualized teaching tactics for students who are struggling in class and offer specialized study resources to them (Boudjehem and Lafifi, 2021). It is in this direction that this study is oriented.

Unfortunately, we did find only in the literature a limited number of conducted works to identify or predict struggling learners using intelligent agents like the works of AL-Malaise et al. (2014) who combined MAS with ML to predict the learners' outcome. We chose to implement a new system combining the advantages of e-learning systems, ITSs, and MASs to identify, predict and intervene with struggling learners.

3.6 Conclusion

The theoretical background that intelligent agents and MASs are built-on is compelling and has much potential. However, putting these rules and concepts into a practical frame needs much effort to understand these ideological bases in the first place and to understand the practical frame allowing their embodiment into profitable products. This chapter tried to gather and analyze existing theories and concepts treating intelligent agents and MAS topics and focus on their utilization in human learning. It also reviewed the effectiveness of using such technology in human learning environments and weight their cognitive, emotional, and motivational effects on the learner.

Unfortunately, we found only a few pertinent works that used intelligent agents and MASs to address the problem of identifying and predicting learning difficulties that learners face within e-learning systems and ITSs. We chose to benefit from the potential and the power of intelligent agents and MASs technologies to treat our study's problem.

In this first part, we presented the theoretical foundations based on the current project is built. The first chapter examined the e-learning and ITS environments, while the second chapter presented a global view of learning difficulties and at-risk and struggling learners. The third chapter provides the basics of DAIs and MASs and their use in education. We saw how psychological and computational approaches are hardly combined. They are, after all, difficult to reconcile.

Nevertheless, our propositions are divided into two categories:

1. The first one is proposing a Learner Model that can be updated in real-time to reflect his behavior. It can also be used to detect and or predict the presence of learning difficulties.
2. The second one is proposing a Multi-Agent based approach to continuously and individually monitor each learner's behavior and try to identify and, or predict scenarios when this learner may face difficulties and take necessary steps to assist and help him.

The type of system that we have chosen to implement has as its main constraints the modeling and monitoring of the learner's behavior to detect or predict situations where a learner may be at-risk or struggling where the chosen system can assist him in overcoming these difficulties.

The set objective is ambitious. Nonetheless, even if our work does not allow us to resolve all of the issues or do so in a thoroughly satisfactory manner, it is worthy of proposing some theoretical and conceptual proposals in this direction and a feasible and implemented solution.

The next part provides all the necessary explanations for our proposed approach to predicting struggling and at-risk learners within a human learning environment that benefits from e-learning systems and ITS. It also utilized a MAS that ensures 24-hours one-to-one monitoring for each learner to identify and predict the presence of learning difficulties.

Part II

Proposed Approach and Results Validation

Chapter 4

Proposed Agent-based Early Warning System (EWS) to Detect and Predict At-Risk Learners

Contents

4.1 Introduction . . . 78
4.2 Problematic of the Research . . . 78
4.3 Contributions of the research . . . 79
4.4 The Proposed Course Structure . . . 81
4.5 Contribution I: Proposition of a New MAS to Detect At-Risk Learners . . . 82
4.5.1 Subsystem 1: The Learner Model Update Subsystem . . . 84
4.5.2 Subsystem 2: The Difficulty Detection Subsystem . . . 86
4.5.3 Subsystem 3: The Intervention Subsystem . . . 95
4.5.4 Intelligent Agents . . . 95
4.6 Contribution II: Extending the proposed MAS to predict At-Risk Learners . . . 97
4.6.1 Subsystem 1: The Learner Model Update Subsystem . . . 98
4.6.2 Subsystem 2: The Difficulty Detection Subsystem . . . 100
4.6.3 Subsystem 3: The Difficulty Prediction Subsystem . . . 100
4.6.4 Subsystem 4: The Intervention Subsystem . . . 103
4.6.5 Intelligent Agents . . . 103
4.7 Proposed Agents Details . . . 104
4.7.1 The Learner Model Update Subsystem: . . . 105
4.7.2 The Difficulty Detection Subsystem: . . . 105
4.7.3 The Prediction Subsystem . . . 107
4.7.4 The Intervention Subsystem . . . 108
4.7.5 Scenario of Learning Difficulties Detection and Prediction Using MAS 109
4.8 Conclusion . . . 110

4.1 Introduction

The teacher's role is to deliver knowledge to his learners. However, doing this using any means, or at any time or pace, would not do the job. It is like throwing objects at someone and expecting him to catch them without considering many variables, such as the weight and speed of the objects or the reflex time needed by that person. Instead, a good teacher would try to have feedback from his learners to assess how they are dealing with the flow of information he keeps sending them. He may use that feedback to adjust the course content or adapt his teaching methods to suit his learners' needs.

Unfortunately, this issue could be more severe in e-learning environments where the learner is alone. He is expected to find the right time, pace, and path to study. Moreover, suppose he faces learning difficulties while following the course in these remote learning environments. In that case, he must also deal with these learning difficulties by himself if we suppose he knows about their existence and nature in the first place. Furthermore, even if he is aware of such difficulties, there is little chance he knows how to deal with them. Unless the learning environment is designed to deal with such scenarios, the result is already sealed, and these learners' fate is already decided.

This chapter presents our proposition of a new agent-based learning environment acting as an Early Warning System (EWS) capable of diagnosing, identifying, and predicting struggling learners and intervening autonomously to assist them. This chapter also presents our two major contributions that constitute the final wanted system. It also provides all the details about each of the final approach's components and each used intelligent agent, and how all of them are collaborating. It describes each conception stage in detail and illustrates it with figures and examples.

4.2 Problematic of the Research

To deal with the problem of the detection and the prediction of learning difficulties faced by learners in online learning environments, the best way to proceed is to design these learning environments to act like human teachers by keeping an eye on each learner individually, monitoring their current and future behavior, and even intervening when necessary to ensure all learners receive their instruction as they should. It is now possible to rely more on the latest Distributed Artificial intelligence advent and breakthroughs like the Multi-Agent Systems to make the design and implementation of such environments possible.

It is in that direction that our research in this project is headed. We aim to cope with the mentioned problems and achieve the expected goals. At this point, we could summarize our research questions in the following:

- How can one identify the most relevant indicators based on the learners' traces that best represent their behavior in e-learning environments?
- How can these indicators be used to detect and predict potential learning difficulties that may be faced by learners in the early stages?
- How can one plan sufficiently early and well-designed interventions based on predictions to support these learners?

- Could one significantly increase struggling learners' cognitive abilities and reduce dropout numbers by proactively supporting these learners?

Before being able to diagnose, detect, and predict any learning difficulties, it is imperative to investigate all possible signs and indicators pointing to the amount and the nature of learning difficulties. The next step is to propose adequate models to reflect the learner's current condition. Using the proper formulas to calculate the level of difficulties each learner faces to assess the impact of such difficulties on that learner and thus understand his behavior. Furthermore, these levels could be used afterward to predict such scenarios for other similar learners, providing valuable time to intervene and change the course of the events before it is late. Intervention could be performed autonomously at early-enough stages to assist learners in overcoming their faced difficulties and enhancing their achievements and performance.

4.3 Contributions of the research

To fulfill these objectives, we proposed a new approach based on intelligent agents capable of modeling, identifying, predicting then assisting learners with difficulties. As shown in Figure. 4.1, the final proposed approach comprises four abstract subsystems that collaborate to model, identify, predict, and intervene to rescue these learners. A "learner model update subsystem" builds and updates learner models. Each of these abstract subsystems delegates a set of intelligent agents to carry out their design tasks.

While the "Difficulty Detection Subsystem" tracks and monitors learners and continues to update a set of primary and secondary indicators and calculate difficulty levels for each "Learning Object" and for each "Course" to estimate their difficulty status. Another "Difficulty Prediction Subsystem" is responsible for predicting possible learning difficulties and those who are likely to face them. Finally, the "Intervention Subsystem" takes care of the intervention requests and takes the necessary measures to assist these learners.

The proposed theoretical approach may seem simple and straightforward. However, in the real world, a learning system has hundreds to thousands of learners and tens to hundreds of courses and teachers. How can one of these proposed subsystems keep track of all the learners and all the courses at the same time? That is why we chose Distributed Artificial Intelligence (DAI) and, more specifically, its most known form, Multi-Agent Systems (MAS). In this direction, each of the proposed subsystems delegates a set of cognitive agents to carry out the tasks it is designed to accomplish. Some agents are assigned to each learner to monitor them individually, while others are assigned to each course.

To propose such an approach, we proceed into two major consecutive and complementary steps proposing one contribution in each step. As a result, we have two major contributions, as shown in Figure. 4.1. The roles and tasks of each abstract subsystem are briefly explained in the following: We provided two major contributions:

Contribution I **At-risk Learners Detection:**
In this contribution, our main objective is to propose a new agent-based approach capable of identifying and assisting struggling learners facing learning difficulties by modeling and analyzing their behavior. We first must find the

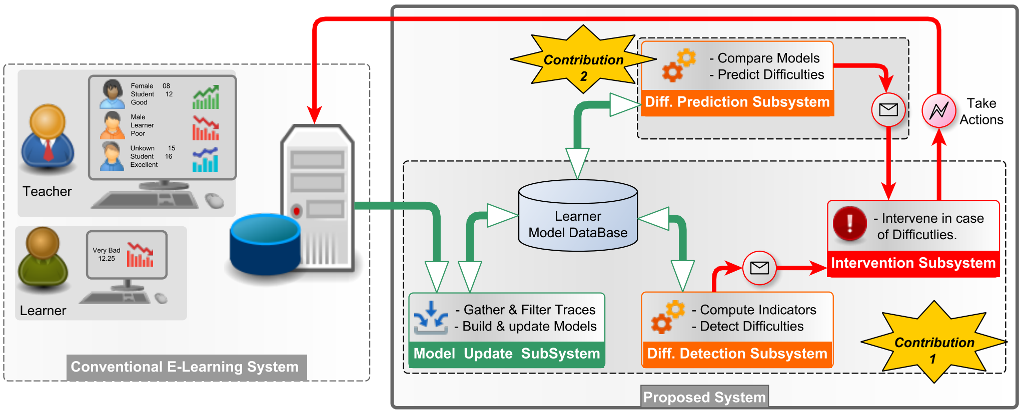


Figure 4.1 – *Proposed Approach to Model, Detect, Predict then Assist Learners*

most relevant indicators reflecting the learner's current condition. Then we need to calculate each of these indicators and combine them to provide a difficulty level for each Learning Object[1] (LO) and course the learner is enrolled in. By identifying the involved learners with each difficulty alongside their location and amount, the proposed approach could intervene autonomously to assist targeted learners. This contribution is divided into the following steps:

(a) Propose a set of computable indicators reflecting the learner's real-time condition.

(b) Propose a formula to calculate a difficulty level for each course and each LO the learner is enrolled in based on the previously mentioned indicators. This level reflects the amount and the location of any difficulty the learner faces.

(c) Propose a "Learner Model" that encapsulates all the difficulties indicators and levels and other necessary information reflecting that learner's condition, activity, and behavior.

(d) Assesses and rates each learner's situation using a color-coded base (Green, Yellow, Orange, Red).

(e) Propose a strategy of intervention to provide the system with action tools to assist and help the learners tagged as having or most like to have difficulties.

(f) Propose a Multi-Agent System that embraces the proposed approach and specifies the role, the nature, and the number of each intelligent agent used.

Contribution II **At-risk Learners Prediction:**

In this second contribution, our main objective is to propose a new agent-based approach capable of predicting the behavior of learners with difficulties at early stages, thus giving more time to assist them, which could help them achieve better.

We first need to update the learner's model to store his learning difficulties history for each LO and course and keep two separate states, "the current state" and "the future state". We could predict the learner's future state by measuring the similarity between his "current state" and the "stored states" of his peers. This contribution is divided into the following steps:

[1] A learning Object is a part of the course (see Subsec. 4.5.1)

(a) Update the "Learner Model" that keeps two separate learning difficulties states, his current and the future one.
(b) Propose a form and strategy to store the learner's history of learning difficulties for each step of the course.
(c) Propose a method to measure the similarity between the active learner's model and the models of his peers.
(d) Build a Prediction Model that uses the Learner's Model and the stored data to predict the future state of the learner.
(e) Update the intervention strategy to cope with predicted learners.
(f) Add new agents to the existing proposed Multi-Agent System to perform the prediction tasks.

The final result is an agent-based system that could fulfill our primary objectives, which are, to model, detect and predict the learners' behavior and foresee any eventual learning difficulties and intervene at early stages to help learners facing these difficulties overcome them. Helping these learners out before it is too late can help instructors and learners improve the learning process.

4.4 The Proposed Course Structure

Before describing the proposed approach, we must first present the structure of the courses and their content.

Each course in the proposition begins on a specific date and stays available till its end date. Each course comprises a set of ordered and chained Learning Objects (LOs). Each LO comprises a set of Learning Activities and Evaluation Activities, as shown in Figure. 4.2.

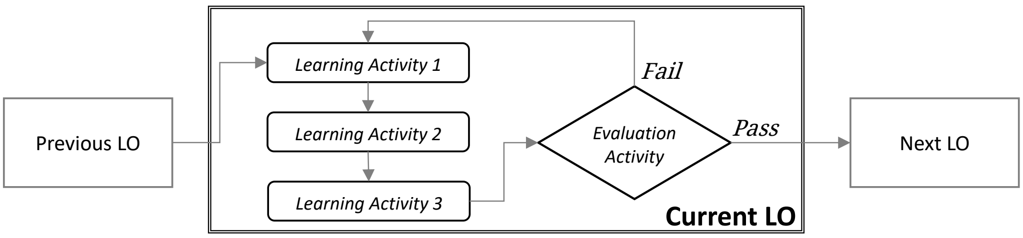


Figure 4.2 – *Example of Learning Object (LO) Composition*

- **Learning Activities:** The main goal of using "Learning Activities" is to let learners acquire information, skills, and competencies by performing specific learning actions on learning objects that are adequate for their content and structure. Each learning activity has to be visited at least once before being considered done. Learning activities could be textual, audio-visual, or a mixture of both.
- **Evaluation Activities:** The primary objective of using "Evaluation Activities" is to determine how much knowledge the learners did assimilate from the "Learning Activities" of a given LO. Evaluation tests like short quizzes or assignments are used as feedback on the quality of acquired knowledge. Each

> Evaluation Activity requires certain conditions before it is accessible to the learner and other conditions to be passed. Each Evaluation Activity has a minimum score to be gained before it is accounted as passed

The LOs in the course are related to each other with a prerequisite relationship (Figure. 4.3). Each LO's Evaluation Activities have to be passed successfully, so the LO is considered "Acquired." The transition to specific LOs requires the acquisition of other dependent LOs, referred to as the "prerequisite" concept. That means the learner cannot access particular LOs unless he acquires and passes all their prerequisite LOs. On the other hand, some LOs may not require the acquisition of any other prerequisite LOs, so they are accessible at any time.

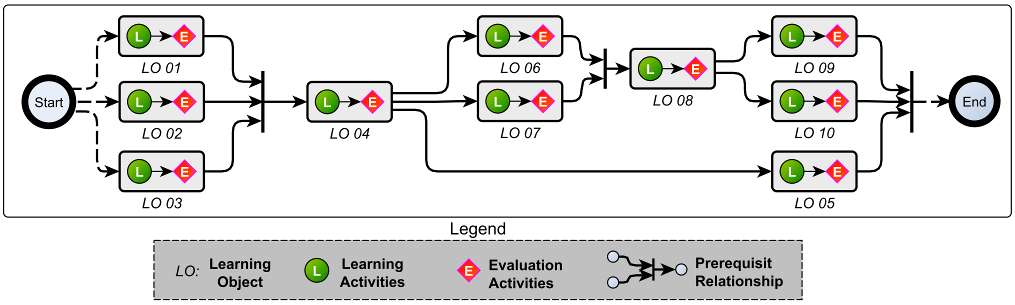


Figure 4.3 – *The Course Structure*

4.5 Contribution I: Proposition of a New MAS to Detect At-Risk Learners

When a learner enrolls in a course, he can access all the available LOs (LOs that do not require any prerequisites and their starting date is due). For each LO, the learner must visit all the "Learning Activities" and then pass their related "Evaluation Activities." However, not all learners will perform the same way. Some of them will pass LOs without blinking, while others will struggle in some of these LOs. The main goal of our proposed approach is to identify and detect those learners. The main idea behind this approach is to closely follow the different changes and fluctuations happening to the learners' achievements to detect if they are facing learning difficulties. By watching the ups and downs of learners' achievements closely, it is possible to see if this learner is improving or struggling. This can be translated graphically and analytically by calculating the slope of the regression of these variations (see Subsec. 4.5.2.1 for more information on the regression slope).

For example, each time a learner accesses an "Evaluation Activity," the scores of all his attempts, best scores, and the number of attempts are stored in his model, whether he passes or fails. Tracking the variation of these scores may come in handy to understand whether he is improving or regressing and can witness whether the Learner is facing learning difficulties or not.

As shown in Figure, we proposed a new approach based on three cooperating subsystems to achieve the abovementioned goals. 4.4. In this contribution, we focus on identifying learners experiencing learning difficulties based on their traces. Identifying these learners could allow assisting them.

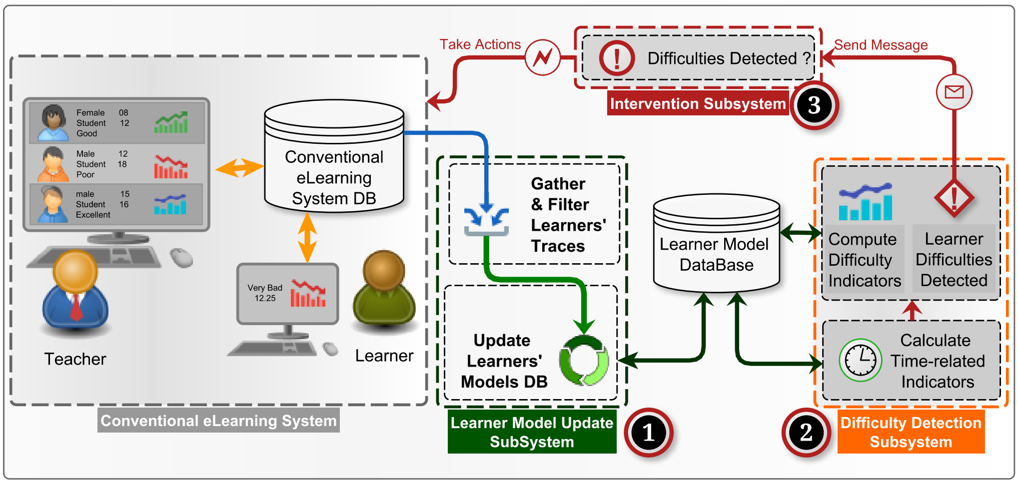


Figure 4.4 – *General Description of A New Approach to Detect Learning Difficulties*

In the beginning, the “Learner Model Update Subsystem” collects and analyzes learners’ traces while interacting within the “Conventional e-Learning System.” It builds for each learner a “Learner Model” containing pertinent information about his activities and behavior. These models are kept up-to-date by the same subsystem reflecting his current condition. The “Learners’ Models” are stored inside the “Learner Model Database” accessible to all the other subsystems.

While the learner carries on his Learning and Evaluation activities, the "Difficulty Detection Subsystem" keep tracking and calculating a set of Primary (see Subsec. 4.5.2.1) and Secondary (see Subsec. 4.5.2.1) indicators and. The same subsystem uses these indicators to calculate a “Difficulty Level” for each LO and each Course of that learner to assess his “Difficulty” condition in real-time. If detected difficulties are above acceptable thresholds, it sends a message to the "Intervention Subsystem," which will intervene on the Learner and takes the appropriate actions

Each of these subsystems has a set of objectives to fulfill as part of the global approach objectives. These objectives are detailed in the following and are explained thoroughly in the next subsections:

1. **The “Learner Model Update Subsystem:”** This subsystem is responsible for perform the following tasks (see Subsec 4.5.1):
   (a) collecting learners left traces while interacting with the “Conventional e-Learning System”.
   (b) building and updating a “Learner Model” for each learner reflecting his current condition. Models are stored inside the “Learner Model Database” accessible to the other subsystems.
2. **The “Difficulty Detection Subsystem"** (see Subsec 4.5.2): it is tasked with:
   (a) calculating a set of indicators using the “Learner Model” stored information.
   (b) calculating a *DifficultyLevel* for each learner for each LO and each course.
   (c) updating the learners’ models with the new calculated indicators and *DifficultyLevels*.

(d) sending a message to the "Intervention Subsystem" in case of any detected learning difficulties. The message specifies the Learner and the LO where the difficulty was detected.

3. **The "Intervention Subsystem"** (see Subsec 4.5.3): This subsystem tasks are:

   (a) keeping the learners and teachers aware of the learning difficulties state.

   (b) notifying the learner and his teachers about any critical difficulties presenting a high level of risk.

4.5.1 Subsystem 1: The Learner Model Update Subsystem

This subsystem is responsible for collecting the learners' traces[2] and then creating and updating a contextual *Model* for each *Learner* present in the "Conventional e-learning System." The other subsystems would use it to detect and assist struggling learners. For example, when the learner enrolls in his first course, the "Learner Model Update Subsystem" creates a model for this learner and stores it in the "Learner Model Database." The model is continuously updated by all the subsystems together.

4.5.1.1 The Course Modelisation

As explained in Subsection 4.4, each *Course* C in our research comprises a set of *LearningObjects LOs* linked between them with a "prerequisite" relationship and having each an evaluation test passed before entering the next *LO*. As a result, we have the following:

- A set of "m" *Learners* $\{L_1, L_2, \cdots, L_i, \cdots, L_m\}$,
- A set of "p" *Courses* $\{C_1, C_2, \cdots, C_j, \cdots, C_p\}$,
- A set of "n" *Learnig Objects (LOs)* for each *Course* C_j=$\{LO_{j1}, LO_{j2}, \cdots, LO_{jk}, \cdots, LO_{jn}\}$.

4.5.1.2 The Learner Modelisation

The *Model* ML_i of the *learner* L_i is defined as $ML_i(S_i, D_i)$, with:

- S_i is the static part of the *Model* ML_i, containing static information like his biographic and registration information.
- D_i is the dynamic part of the *Model* ML_i containing all cognitive and behavioral data about the learner L_i for each enrolled *Course* C_j (Figure. 4.5).

[2] Learner Traces or Learner "digital breadcrumbs" refer to data generated by the interaction of a user with a given system going from simple mouse clicks to whole psychological and sociological profiles. There is a rising interest in the automatic analysis of such data because researchers are confident that it can inform us about learners' behaviors and help us improve learning experiences (Brahim and Lotfi, 2020). They represent the source of most educational data (Clemens et al., 2018)

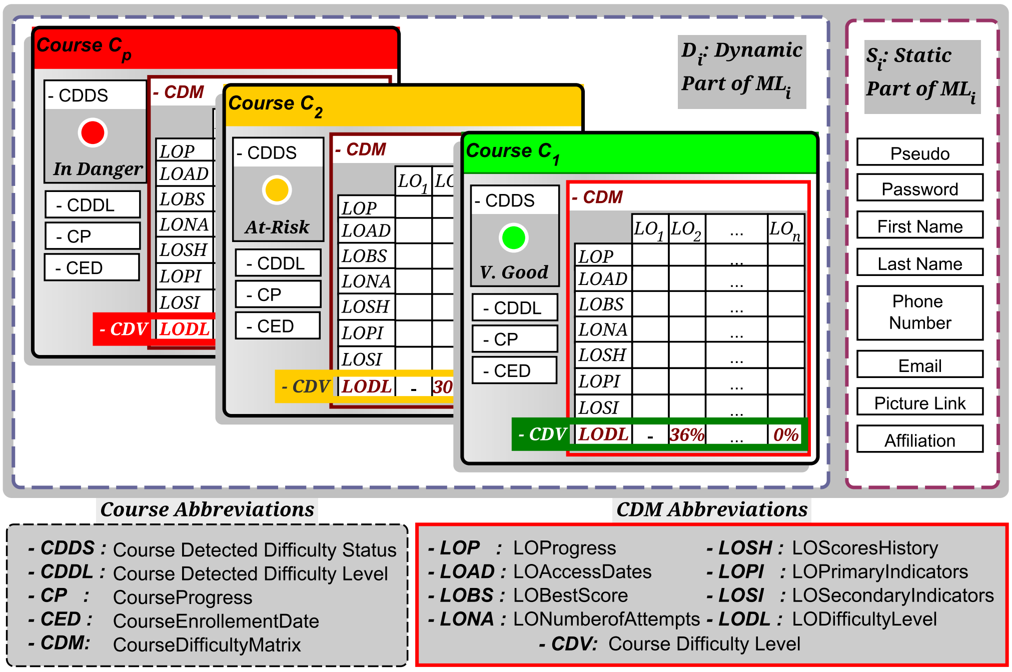


Figure 4.5 – *ML_i Model of the Learner L_i*

$$D_i = \bigcup_{j=1}^{p} C_{ij} \tag{4.1}$$

Formula. 4.1 defines D_i, where C_{ij} is the *Course* C_j enrolled by the *Learner* L_i and p is the number of the enrolled courses.

Furthermore, $C_{ij} = \{CP_{ij}, CDDS_{ij}, CDDL_{ij}, CP_{ij}, CED_{ij}, CDM_{ij}\}$, where:

$CDDS_{ij}$: *Course* C_j Difficulty Status of the *Learner* L_i.

$CDDL_{ij}$: *Course* C_j Difficulty Level of the *Learner* L_i.

CP_{ij} : *Course* C_j Progress of the *Learner* L_i.

CED_{ij} : *Course* C_j Enrollment Date of the *Learner* L_i.

CDM_{ij} : *Course* C_j Difficulties Matrix of the *Learner* L_i.

$$CDM_{ij} = \bigcup_{k=1}^{n} LO_{ijk} \tag{4.2}$$

Formula. 4.2 defines the *CourseDifficultiesMatrix* CDM_{ij}, which contains all calculated variables for the *Learner* L_i towards the *Course* C_js, and n is the number of the started *LO*s.

$LO_{ijk} = \{LOP_{ijk}, LOAD_{ijk}, LOBS_{ijk}, LONA_{ijk}, LOSH_{ijk}, LOPI_{ijk}, LOSI_{ijk}, LODL_{ijk}\}$, where:

LOP_{ijk} : Progress of the *Learner* L_i on the LO_k of the *Course* C_j.

$LOAD_{ijk}$: LO_k Access Dates of the *Learner* L_i on the LO_k of the *Course* C_j.

$LOBS_{ijk}$: LO_k Best Score of the *Learner* L_i on the LO_k of the *Course* C_j.

$LONA_{ijk}$: LO_k Number of Attempts of the *Learner* L_i to pass the evaluation test of LO_k of the *Course* C_j.

$LOSH_{ijk}$: LO_k Scores History of the *Learner* L_i on the LO_k of the *Course* C_j.

$LOPI_{ijk}$: LO_k Primary Indicators values of the *Learner* L_i on the LO_k of the *Course* C_j (see Subsec 4.5.2).

$LOSI_{ijk}$: LO_k Secondary Indicators values of the *Learner* L_i on the LO_k of the *Course* C_j (see Subsec 4.5.2).

$LODL_{ijk}$: LO_k Difficulty Level of the *Learner* L_i on the LO_k of the *Course* C_j.

4.5.2 Subsystem 2: The Difficulty Detection Subsystem

This subsystem is responsible for assessing the condition of each learner in all his enrolled courses. For that, it uses two main variables: *CourseDifficultyLevel* (CDDL) and *CourseDifficultyStatus* (CDDS).

CDDL is a percentage value that reflects how many difficulties this *Learner* is facing during this *Course*. On the other hand, CDDS is a scalar value used to classify the *Learner* in one of four states (see Formula. 4.3).

CDDL and CDDS are global values calculated for each course. For example, a CDDL value of 100% means the *Learner* is "in Danger," while 0% means that he is "Very Good." Four Learner states are used and stored in the CDDS by comparing CDDL to a predefined set of static thresholds. CDDS value is defined as one of four states, each identified visually by color (see Formula. 4.3).

$$CDDS_{ij} = \begin{cases} \textbf{Very Good} \text{ ("Green")}, & \text{if}(CDDL_{ij} \geq 75\%) \\ \textbf{Good} \text{ ("Yellow")} & \text{if}(75\% > CDDL_{ij} \geq 50\%) \\ \textbf{At-Risk} \text{ ("Orange")}, & \text{if}(50\% > CDDL_{ij} \geq 25\%) \\ \textbf{In Danger} \text{ ("Red")}, & \text{if}(CDDL_{ij} < 25\%) \end{cases} \tag{4.3}$$

The CDDL of the *Course* is the mean value of the LODLs of all available *LO*s that compose this course. So, to calculate the CDDL of the Learner L_i in the Course C_j $CDDL_{ij}$ we used all stored $LODL_{ijk}$ (Formula. 4.4).

$$CDDL_{ij} = \frac{\sum_{k=1}^{n} LODL_{ijk}}{n} \tag{4.4}$$

It is essential to highlight here that even though the LODLs are calculated for each *LO* separately, some indicators may need information about previous *LOs*. The *LearningObjectDifficultyLevel* (LODL) is calculated on the basis of a set of indicators grouped according their impact into *Primary* and *Secondary* indicators. *Primary* indicators has a "high impact" (HI) on calculating the LODL while *Secondary* indicators has a "low impact" (LI) on LODL value. Moreover, in the following list, we named these indicators in correspondence with their order as follows:

A. **Primary Indicators** are critical. They have a "High Impact" on the value of LODL. The existence of each one of them would increase the LODL value with the "HighImpact" value. By default, HI equals 2, but the course designer can modify it. These indicators are :

   1. **HI1**: Maximum Number of evaluation's Attempts Reached (***MNAR***) (see Subsection. 4.5.2.1).
   2. **HI2**: The Slope of regression of the Past three passed *LOs'* evaluations Best Scores (***SPLOBS***) (see Subsection. 4.5.2.1).
   3. **HI3**: The Slope of the Last three Attempts' scores of the Current *LO*'s evaluation (***SLACLO***) (see Subsection. 4.5.2.1).

B. **Secondary Indicators** have a "Low Impact" on the value of LODL. The existence of each one of them would increase the LODL value with the "LowWeight" value. By default, LI equals 1, but the course designer can modify it. These indicators are:

   1. **LI1**: The number of Delay Days past *LO* deadline (***NDDLO***) (see Subsection. 4.5.2.1).
   2. **LI2**: The Slope of the regression of the Number of Attempts to pass the Past three *LOs'* evaluation tests (***SNAPLO***) (see Subsection. 4.5.2.2).

Each *LO* has a start date and a deadline date. Learners cannot access LOs before they start, and they have to access the *LO* before the deadline is reached. Once the *LO* starts we have two cases (see Figure. 4.6):

1. **Case 2:** either the Learner accesses the *LO* before its deadline is reached.
2. **Case 1:** or *LO* deadline is reached before the learner access it.

In either case, "The Difficulty Detection Subsystem" creates a new record in the learner model corresponding to that *LO* with a LODL value of zero.

Case 1: the Learner accessed the *LO* before its deadline is reached

Whenever a significant change in one of the indicators is detected, "The Difficulty Detection Subsystem" recalculates the LODL accordingly by adding one of two values: *HighImpact* (HI) for *Primary* indicators and *LowImpact* (LI) for *Secondary* ones (Figure. 4.6). For example, if the third primary SPLOBS indicator is lower than -0.25, $HI_3 = 1$.

As we previously explained, HI and LI are assigned values 2 and 1 by default in the system and could be adjusted by the *Course Teacher* (HI has to be greater or equal to LI). As a result, the maximum value for LODL is eight

$$\sum_{i=1}^{3} HI_i + \sum_{j=1}^{2} LI_j = 3 * 2 + 2 * 1 = 8$$

when all indicators are present. Nevertheless, we want the value of LODL as a percentage, so we divide the calculated LODL value over 8 (Formula 4.5).

$$LODL = \frac{\sum_{a=1}^{3} HI_a + \sum_{b=1}^{2} LI_b}{8} \tag{4.5}$$

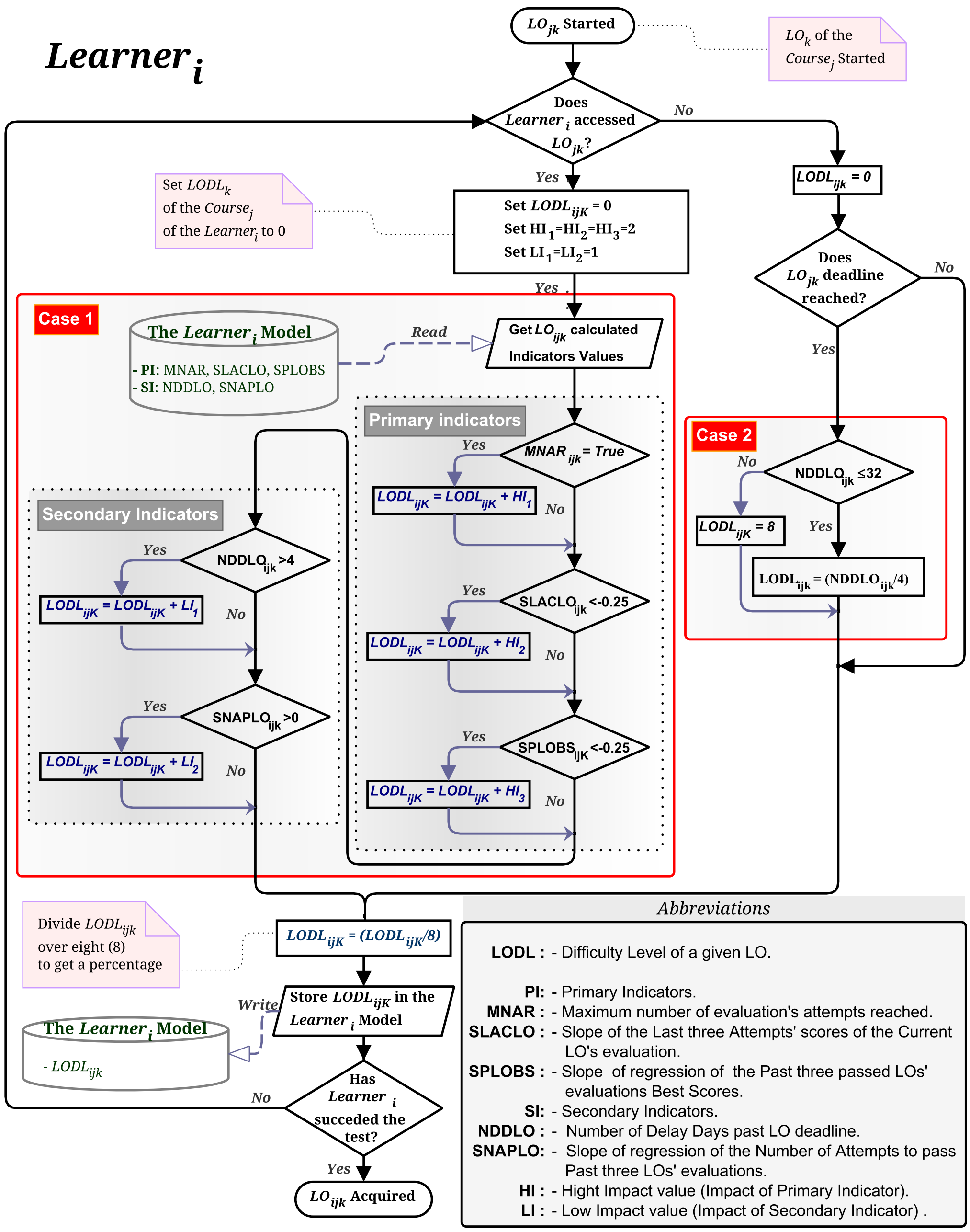


Figure 4.6 – *Flow Diagram of the calculation algorithm of $LODL_{ijk}$ of the LO_k of the Course C_j of the Learner L_i, based on Primary and Secondary indicators*

.

Case 2: Learner did not access an *LO* that reached its deadline

If the learner did not access a given LO in time, there is no way to follow his condition because he does not achieve anything yet in that LO. However, when this LO reaches its deadline, and the learner still does not access it, there may be a problem with that learner, and a danger flag must be raised. In that case, the number of delay days will significantly affect the $LODL_{ijk}$ value as it is the only available. When the learner accesses the LO, the LODL is recalculated using all indicators, as explained in case 1.

4.5.2.1 Primary and Secondary Difficulty indicators

At this point, we thoroughly explain the previously mentioned Primary and Secondary indicators. Most of our used indicators are based on the value of the slope of the regression. As a result, it is imperative to define the meaning of that slope and how it can be calculated. This subsection explains how it can be calculated.

What is a Slope of regression? In statistics, to model the relationship between a scalar response (dependent variable) and one explanatory variable (independent variable), we use a linear approach called "simple Linear Regression". The slope (m) of a line denotes its steepness, whereas the intercept (b) reveals where it intersects the Y-axis. The slope and intercept are used to assess the average rate of change in a linear connection between two variables (Figure. 4.7). In our situation, each chosen variable

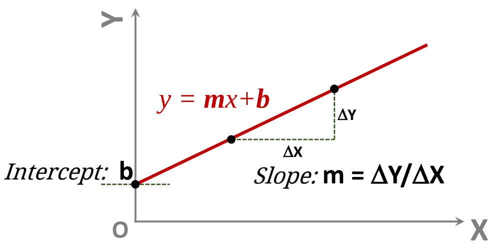


Figure 4.7 – *Simple Linear Regression Slope and Intercept*

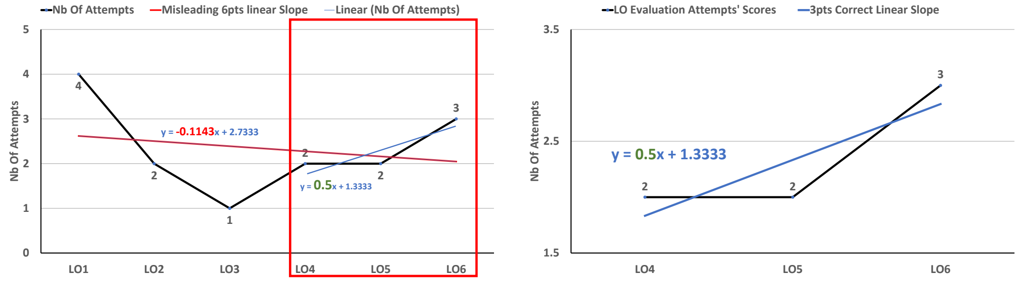
value (a score value, for example) could be seen as the y coordinate of a geometric point and the attempt number (or LO number) as its x coordinate. We have to find the Least Squares Regression Line slope: $y = mx + b$ that is the closest to these points, where m is the slope value and b is the intercept. We want to minimize the sum of the squares of the vertical distances. That is finding m and b such that ${d_1}^2 + {d_2}^2 + {d_3}^2$ is minimum. To calculate the slope m we use Formula 4.6.

$$m = \frac{n\sum_{i=1}^{n}(x_i y_i) - \sum_{i=1}^{n}(x_i)\sum_{i=1}^{n}(y_i)}{\sum_{i=1}^{n}(x_i)^2 - (\sum_{i=1}^{n}(x_i))^2} \tag{4.6}$$

Why taking only the last three Scores/Attempts? Most of our indicators are calculated based only on the last three measured values, which is for a good reason. Computing the slope of the whole history of a given variable may not give meaningful information that can help us understand the real-time situation of the learner. Using the totality of the values may give misleading information. For example, computing the slope of the number of attempts for all six (06) LO gives a negative result (Figure 4.8(a)), while visually, we can see that this number was decreasing for the three first LOs than starts to increase. So visually, we can see that the learner is doing good now, while the computed slope indicates the inverse. In contrast, if we compute the slope for only the last three LOs, we will get a positive slope (Figure 4.8(b)).

A. Primary Indicators

As mentioned before, these indicators are of utmost importance and play a crucial role in determining the difficulties severity.



(a) (b)

Figure 4.8 – *(a) Computed slope of regression using six LO points, (b) Computed slope of regression using three LO points*

1. Maximum Number of evaluation Attempts Reached (MNAR): The Learner is allowed to retry an assessment several times until he succeeds. The teacher can fix the maximum number of attempts for each "Evaluation Activity" or leave it by default. If a learner attempts an "Evaluation Activity" more times than allowed before completing it, there is a good chance that he is having difficulty passing it. When this happens, the MNAR indicator value becomes "True," otherwise, its default value is "False." Even though the learner reaches the limit, if he does not pass the "Evaluation Activity," he can reattempt the evaluation until passing it.

As shown in Figure. 4.6, if this indicator is true the value of $LODL_{ijk}$ is increased by HI_1 (high impact value which has a value of "2" by default).

$$\textbf{IF } (MNAR_{ijk} = True) \textbf{ Then } LODL_{ijk} = LODL_{ijk} + HI_1;$$

2. The Slope of the regression of the Past three passed *LOs*'s evaluations Best Scores (SPLOBS): As the Learner progress through the Course, he must pass the evaluation tests for the previous *LOs* before passing on to the next ones. The best score for each *LO* is stored in the Learner Model. The value of this indicator for the current *LO* is the regression slope of the best scores of the past three *LOs*. A positive slope indicates that the Learner is finding his way through the Course and that things are getting easier for him as he progresses through the Course, while a negative slope indicates that the Course is starting to become more difficult for him. A zero slope implies stagnation, so there has been no significant positive or negative change (see Example 2).

How does SPLOBS affect the LODL? As shown in Figure. 4.6, if $SPLOBS_{ijk} < -0.25$, $LODL_{ijk}$ is increased by HI_2 (high impact value which has a value of "2" by default) using the following condition:

$$\textbf{IF } (SPLOBS_{ijk} < -0.25) \textbf{ Then } LODL_{ijk} = LODL_{ijk} + HI_2;$$

This value "-0.25" is chosen based on our preliminary tests. Example 2 provides more details about this indicator calculation.

Example 2. For a learner to pass an LO's "Evaluation Activity," he has to get a score above the minimum. Each time he retries "Evaluation Activities," his best scores are stored in his model. Table. 4.1 represents an example of a succession of the best scores of a learner during a course, where each column represents his best score obtained for an *LO*'s evaluation test.

LO Nb.	*LO1*	*LO2*	*LO3*	*LO4*	*LO5*	*LO6*
Best score (/100)	**55**	**50**	**80**	60	50	
SPLOBS				**12,5**	0	-20

Table 4.1 – *Example of the SPLOBS of a given learner*

To calculate the $LODL_{ij4}$, we need to calculate the slope of the regression of the past three LOS of LO_4, $SPLOBS_{ij4}$, which are LO_1, LO_2, and LO_3 best scores to see if the learner's best scores are increasing or decreasing. In the case of As shown in (Figure. 4.9), $SPLOBS_{ij4}$=12.5.

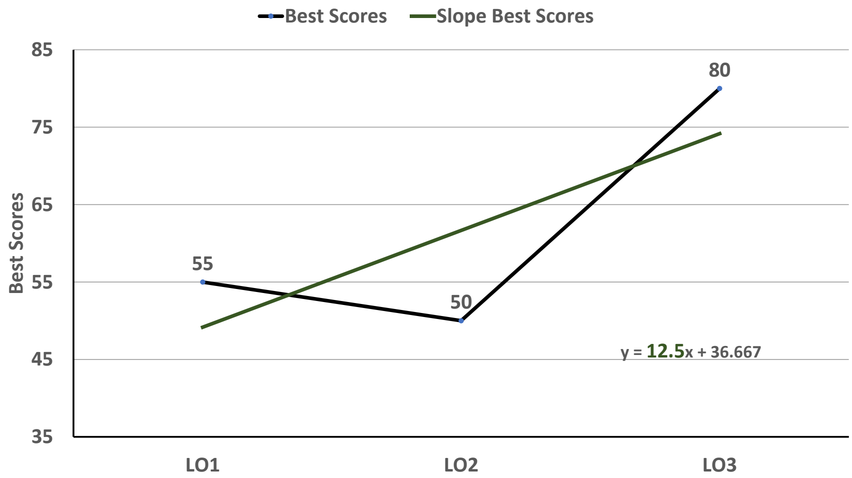


Figure 4.9 – *Example of SPLOBS (Improvement)*

As we can see in Table. 4.1, graphically, we could have three cases:

- **Improvement** : $SPLOBS_{ij4}$=12.5 as we have *LO1*(55), *LO2*(50), *LO3*(80), the slope for (Figure. 4.9). The condition $SPLOBS_{ijk} < -0.25$ is not met the value of $LODL_{ijk}$ does not change
- **Stagnation** : $SPLOBS_{ij5}$=0 as we have *LO2*(50), *LO3*(80), *LO4*(60), the slope for *LO5* is 0. Same as before, The condition $SPLOBS_{ijk} < -0.25$ is not met the value of $LODL_{ijk}$ does not change.
- **Regression** : $SPLOBS_{ij4}$=-20 as we have *LO3*(80), *LO4*(60), *LO5*(50), the slope for *LO6* is -20. In this case, the condition $SPLOBS_{ijk} < -0.25$ is met, $LODL_{ijk} = LODL_{ijk} + HI_2$.

3. The slope of the Last three Attempts' scores of the Current *LO*'s evaluation (SLACLO): The learner is allowed to repeat the same evaluation test several times until he succeeds. Each time the learner fails an assessment, his attempt score is stored in his Model. Tracking the variation of these scores can be used to understand whether this learner understands this *LO's* concepts well or not. This indicator represents the Slope of the regression of this *LO's* last three attempts on the given assessment.

A positive value of this indicator means that the Learner's scores keep improving, while a negative value indicates that the test scores worsen over time. A zero slope indicates stagnation, so there is no improvement or regression.

How does SLACLO affect the LODL? As shown in Figure. 4.6, if $SLACLO_{ijk} < -0.25$, $LODL_{ijk}$ is increased by HI_3 (high impact value which has a value of "2" by default) using the following condition:

$$\textbf{IF } (SLACLO_{ijk} < -0.25) \textbf{ Then } LODL_{ijk} = LODL_{ijk} + HI_3;$$

This value "-0.25" is chosen based on our preliminary tests. Example 3provides more explanations.

Example 3. During his first two attempts, the *Learner* L_i repeats the LO_{jk} evaluation test several times before he can pass it. His attempts' scores are stored in his model. In the following, we present three examples of consecutive obtained scores during these attempts that represent three slope states:

- **Improvement** : If the learner got scores of 30, 45, then 55 over 100, respectively. $SLACLO_{ijk} = 12.5$. in this case, the condition $SLACLO_{ijk} = 12.5 < -0.25$ is not met. The *Learner* L_i attempts' scores are improving, it is more likely that this learner is not facing difficulty, and $LODL_{ijk}$ keep its original value;
- **Stagnation** : If the learner got scores of 45, 30, and 45, respectively. $SLACLO_{ijk} = 0$. in this case, the condition $SLACLO_{ijk} = 12.5 < -0.25$ is also not met. The *Learner* L_i attempts' scores are stagnating (not improving nor regressing), it is more likely that this learner is not facing difficulty and $LODL_{ijk}$ keep its original value; so, no improvement nor regression.
- **Regression** : If the learner got scores of 40, 30, and 25, respectively. $SLACLO_{ijk} = -7.5$. in this case, the condition $SLACLO_{ijk} = 12.5 < -0.25$ is met. The *Learner* L_i attempts' scores are degrading, it is more likely that this learner is facing difficulty and $LODL_{ijk} = LODL_{ijk} + HI_2$ (Figure. 4.10).

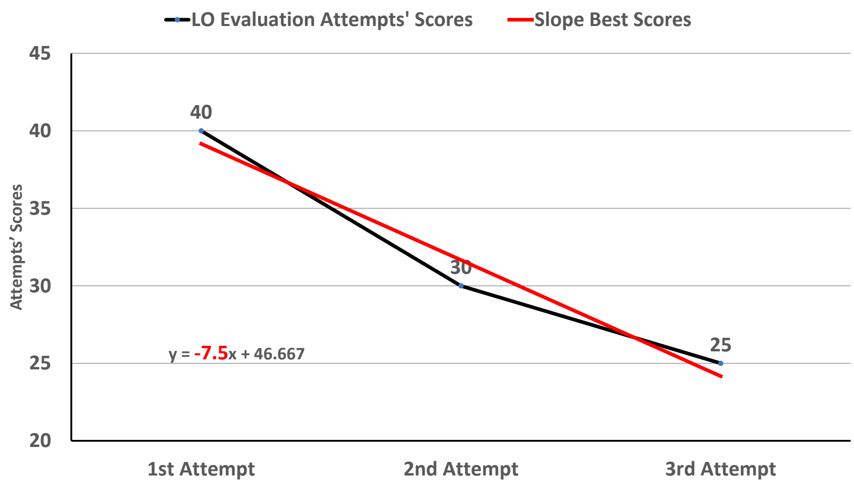


Figure 4.10 – *Example of SLACLO (Regression)*

B. Secondary Indicators

As their names indicate, these indicators are of less importance. They are used to fine-tune the difficulty levels.

1. Number of Delay Days past *LO* deadline (NDDLO): Each *LO* has a StartDate and a Deadline (expressed in days number). The longer it takes for a learner to access a specific started LO, the more likely this learner is to face difficulties. This indicator is re-evaluated daily for all started *LOs* from all courses and for all enrolled learners.

As shown in Figure. 4.11, LO_{jk} is not accessible before the StartDate.

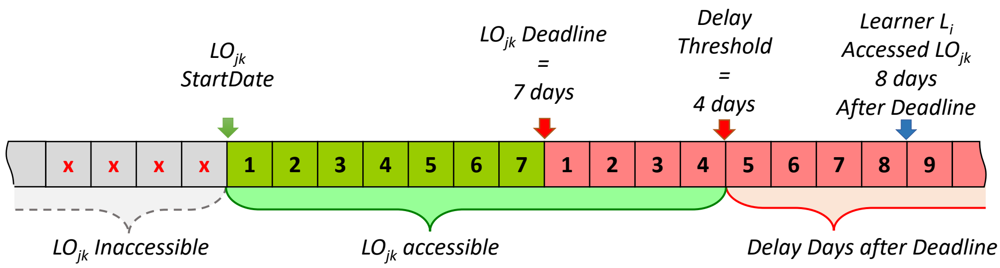


Figure 4.11 – *Number of Delay Days of Learner L_i accessing a Learning Object LO_{jk} past Deadline*

The Deadline is the number of days that *Learner* L_i should access LO_{jk}. When the *Learner* L_i accessed the LO_{jk} after the Delay Threshold period with eight (8) days from the Deadline, for example, he is considered as late, and the number of days after the deadline is accounted for every day until he finally accesses that LO_{jk}.

The value of the NDDLO indicator increases every day and will only permanently freeze when the learner finally accesses the *LO* concerned.

4.5.2.2 How does NDDLO affect the LODL?

The NDDLO can affect LODL in two ways depending on whether the learner accessed or not the corresponding LO before the deadline is reached.

Case 1: the *Learner* L_i did access the LO_k of the *Course* C_j : In this case, it does not matter if the learner accessed the LO before or after the deadline. If the learner accessed the LO before the deadline is reached, the value of $NDDLO_{ijk} = 0$ and the $LODL_{ijk}$ is not affected by this indicator. However, if the learner has accessed the LO after the deadline of more than four (4) days, the $LODL_{ijk}$ is increased with LI_1 which has a default value of "1" (Figure. 4.12).

$$\textbf{IF } (NDDLO_{ijk} > 4) \textbf{ Then } LODL_{ijk} = LODL_{ijk} + LI_1;$$

Case 2: the *Learner* L_i did not access the LO_k of the *Course* C_j after the deadline was reached: In that case, we do not have any information about that learner apart from the fact that he did not access the LO in time, meaning whether he could not acquire the prerequisite LOs or had another difficulty. In this case, the number of delay days is the only way to calculate the $LODL_{ijk}$ as it is the only available indicator to raise the alarm of potential difficulty (Figure. 4.13).

$$\textbf{IF } (NDDLO_{ijk} <= 32) \textbf{ Then } LODL_{ijk} = (NDDLO_{ijk}/4);$$
$$\textbf{Else } LODL_{ijk} = 8;$$

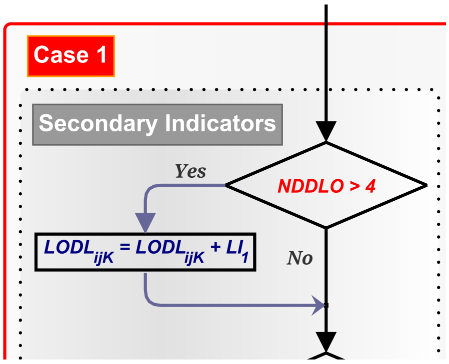


Figure 4.12 – *Case 1: The Learner L_i did access the LO_k of the Course C_j*

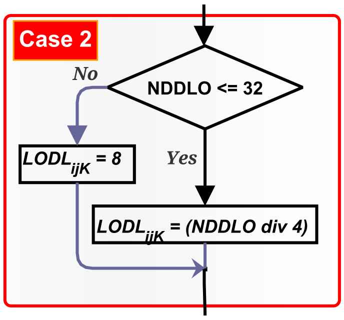


Figure 4.13 – *Case 2: The Learner L_i did not access the LO_k of the Course C_j after the deadline*

2. The slope of the regression of the Number of Attempts to pass the Past three *LOs'* evaluation tests (SNAPLO)

This indicator tracks the number of attempts taken by the learner before completing the *LO* assessments successfully. For each unsuccessful *LO* assessment attempt, the number of attempts is stored in the Learner's Model. This indicator tracks the Slope of the number of shots taken before passing the last three *LOs*.

Positive values of this indicator imply that the learner is finding it increasingly difficult to take the evaluation tests, causing him to attempt the tests more often than last time.

In contrast, negative values are a good sign indicating that the learner is doing better. As with the previous indicator, there are many other scenarios where the learner is likely to repeat the tests more than once

How does SNAPLO affect the LODL? As shown in Figure. 4.6, if $SNAPLO_{ijk} > 0$, $LODL_{ijk}$ is increased by LI_2 (Low impact value which has a value of "1" by default) using the following condition:

$$\textbf{IF } (SNAPLOijk > 0) \textbf{ Then } LODL_{ijk} = LODL_{ijk} + LI_2;$$

This value "0" is chosen based on our preliminary tests. Example 4provides more explanations.

Example 4. In *LO1*, the learner attempted the evaluation test three times before passing on the fourth one. Then, two times in *LO2* and three times in *LO3*. To calculate this indicator for *LO4*, we use the information from *LO1*, *LO2*, and *LO3*. The calculated slope for (2, 2, 3) is 0.5. However, in this case, a positive value means that the Learner is making more and more attempts before passing the assessment test, which means that he is struggling (Figure. 4.14).

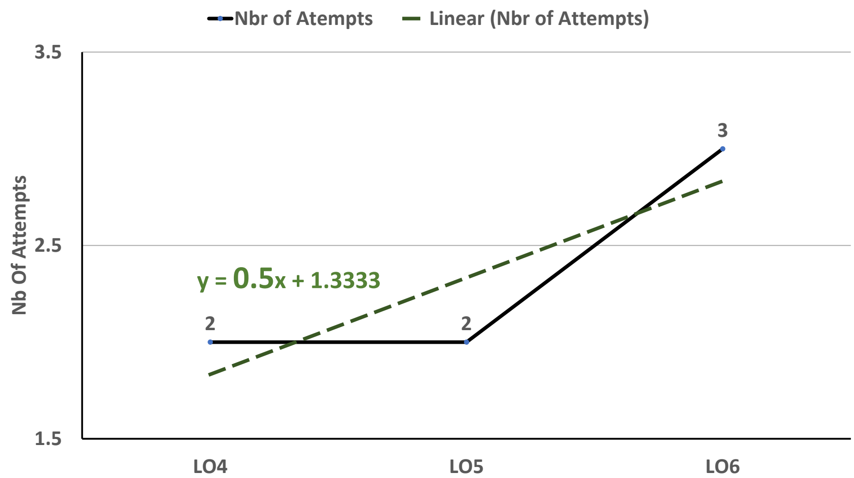


Figure 4.14 – *Example of SNAPLO.*

4.5.3 Subsystem 3: The Intervention Subsystem

It is responsible for taking the necessary actions to save the learners, either by contacting the Learner himself or by warning another actor that can intervene with the learner, like the teacher. It pushes system notifications to learners and teachers about any probably faced critical situation. For the learner, this subsystem displays the learner's current difficulty.

On the other hand, the teacher receives weekly reports about his learners' achievements, progress, and, more importantly, about learners identified as "at-risk." Furthermore, it also sends emails and SMSs to learners and teachers, thus ensuring that all the actors are informed even if they do not enter the e-learning system.

4.5.4 Intelligent Agents

After explaining the proposed approach as a system comprising a set of abstract subsystems, this subsection explains the use of intelligent agents within the proposed approach.

The proposed approach comprises intelligent and autonomous subsystems collaborating to accomplish the system's main goals with minimum human intervention. However, the number of learners and courses is enormous compared to the number of subsystems, which could make the system more sequential-like and less responsive. That is why each of these subsystems is powered by a set of intelligent agents undertaking the tasks for each learner and each course individually.

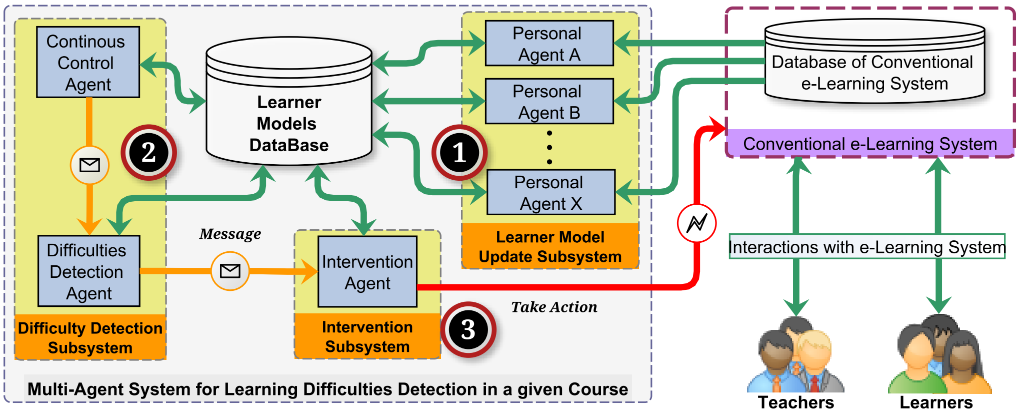


Figure 4.15 – *Proposed Multi-Agent System for Learning Difficulties Detection in a given Course*

As shown in Figure. 4.15, four types of agents are used.

The four used agents are distributed as follows:

A. **The Learner Model Update Subsystem:** uses one type of agent:

1. **PersonalAgent:** This agent creates and then keeps updating the contextual Model of the Learner. One instance of this agent is associated with each Learner, and each time the Learner access the System, his PersonalAgent is activated and put to work.

B. **The Difficulties Detection Subsystem:** rely on two types of agents:

1. **ContinuousControlAgent:** This agent performs all time-related tasks and checks the learners' persistence. One instance of this agent is associated with each course. This agent is triggered one time each day.
2. **DifficultyDetectionAgent:** This agent calculates the CourseDifficultyLevel of every Learner in real-time. In case of any detected difficulties, it notifies the intervention agent of any noticed difficulties. One instance of this agent is associated with each course. This agent is triggered on the first day of the course and discarded on the last day.

C. **The Intervention Subsystem:** needs one type of agent:

1. **InterventionAgent:** The role of this agent is to take action to save detected or predicted in-distress learners. One instance of this agent is associated with each course.

Section. 4.7 provides all the details about the implemented agents.

4.6 Contribution II: Extending the proposed MAS to predict At-Risk Learners

In this contribution, we proposed a new approach of an early warning system (EWS) capable of predicting as soon as possible learners who are facing problems and able to intervene autonomously to assist them before it is too late.

The prediction is based on information gathered and stored in the first contribution, constituted essentially by the history of difficulties of each learner (see Section. 4.5). By seeking similarities between the current learner's known history of difficulties and that of his predecessor peers, we could predict the behavior of that learner in a given course.

To perform the prediction process, we continued upon our previous contribution. We added a new "*Difficulty Prediction Subsystem*" responsible for the prediction operation. However, the three subsystems from our previous contribution did not get many updates except for the "Learner Model," which received some modifications. With the addition of the fourth module, the proposed approach can perform two functions simultaneously: detecting and predicting the behavior of learners with difficulties.

Figure 4.16 presents the function of the new proposed subsystem as it collaborates with the other subsystems. In the beginning, when a learner subscribes to the system, the "*Learner Model Update Subsystem*" starts by creating his Model and keeps it updated using gathered traces.

The "*Difficulty Detection Subsystem*" uses the Model to calculate and update the learner's *CourseDetectedDifficultyLevel* (CDDL) towards each enrolled course based on all its constituent LearningObjects' (LOs) estimated *LearningObjectDifficultyLevels (LODLs)* calculated by their turn based on a set of Primary and Secondary Indicators. This CDDL is a quantitative representation of the level of difficulties faced by the learner and is used to estimate his *CourseDetectedDifficultyStatus* (CDDS) towards that course. Each range of CDDL values corresponds to a state of CDDS.

All Calculated LODLs of each enrolled *Course* by a *Learner* L_i constitute a *CourseDifficultyVector* (CDV) for that learner stored in his *Model* and used by the "*Difficulty Prediction Subsystem*" to calculate his *CourseDropoutProbability* (CDP) and *CoursePredictedStatus* (CPS) (see Subsection 4.6.3). In case of any detected or predicted problems for a learner in a given course, the "*Intervention Subsystem*" is contacted to take the necessary actions.

The proposed approach now comprises four cooperating and communicating subsystems:

1. **The "Learner Model Update Subsystem:"** this subsystem has already been explained in subsection 4.5.1; however, it has been updated to take into consideration the prediction tasks. All the new modifications are thoroughly explained in subsection 4.6.1.
2. **The "Difficulties Detection Subsystem"** a full explanation on that subsystem is already provided in subsection 4.5.2.

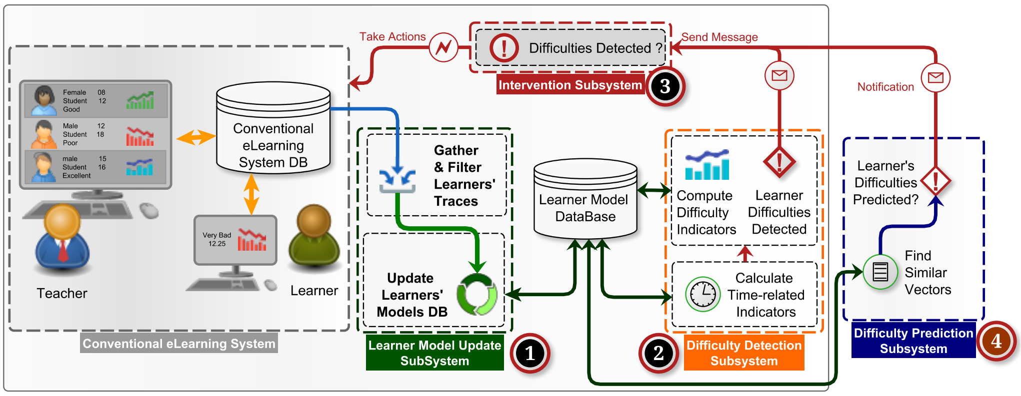


Figure 4.16 – *General Description of the Approach to Detect and Predict Learning Difficulties*

3. **The "Intervention Subsystem"** This subsystem also has been explained in subsection 4.5.3
4. **The "Difficulties Prediction Subsystem"** this subsystem details are in provide in subsection 4.6.4. for each course and each learner, this subsystem:
   (a) continuously seeks similarities between the current learner's CDV and those of his predecessor peers.
   (b) estimates a *CoursePredictedStatus* (CPS) based on calculated a *CourseDropoutProbability* (CDP) to predict whether this learner will drop out or not.
   (c) updates the learners' models with the new calculated CPSs and CDPs.
   (d) sends a message to the "Intervention Subsystem" in case of any difficulties Predicted. The message specifies the Learner and the course where the difficulty was predicted.

4.6.1 Subsystem 1: The Learner Model Update Subsystem (Updated from Contribution 1)

In this second contribution, we updated the "Learner Model" to consider the Learner Predicted Status. As shown in Figure 4.17, the *Course* representation is now defined as: In addition to the current detected status, the Course in the model has a predicted future status. $C_{ij} = \{CPS_{ij}, CDP_{ij}, CDDL_{ij}, CDDS_{ij}, CDV_{ij}, CP_{ij}, CED_{ij}\}, where :$

CPS_{ij} : *Course* C_j Predicted Status of the *Learner* L_i.

CDP_{ij} : *Course* C_j Dropout Probability of the *Learner* L_i.

$CDDL_{ij}$: *Course* C_j Detected Difficulty Level of the *Learner* L_i.

$CDDS_{ij}$: *Course* C_j Detected Difficulty Status of the *Learner* L_i.

CDV_{ij} : *Course* C_j Difficulty Vector of the *Learner* L_i.

CP_{ij} : *Course* C_j Progress of the *Learner* L_i.

CED_{ij} : *Course* C_j Enrollment Date of the *Learner* L_i.

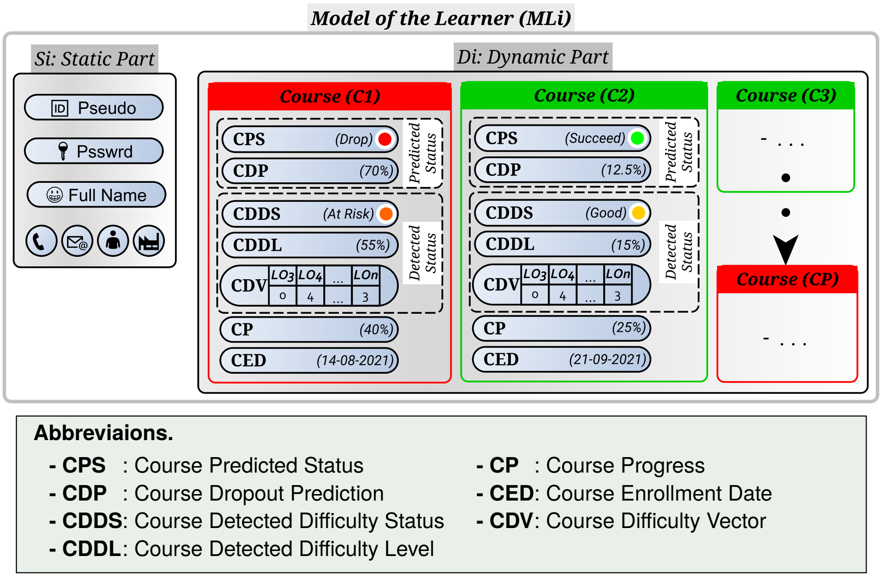


Figure 4.17 – *New Model of The Learner Taking into Consideration the Prediction*

When a learner enrolls in his first Course, the "*Learner Model Update Subsystem*" creates a model inside the "*Learner Model Database*" and stores important information using various variables like his *CourseProgress* (CP) or his *CourseEnrolmentDates* (CED). Whenever the learner advances in the Course, The "*Difficulty Detection Subsystem*" and the "*Difficulty Prediction Subsystem*" keep updating other variables in that learners' Model (See Subsection 4.6.2 and Subsection 4.6.3).

The CDV_{ij} of the *Learner* L_i in the *Course* C_j is a numeric vector constantly updated by the "*Difficulty Detection Subsystem*" and contains all his difficulty history (see Subsection 4.6.2).

When *Learner* L_i passes the LO_k, his CDV_{ij} is at Level k and contains only "k-known LO" Detected Difficulty (LODDs) while the other LODDs beyond LO_k remain unknown (see the example presented on Figure 4.18).

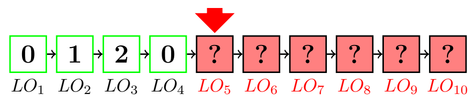


Figure 4.18 – *Learner L_i CDV after passing LO_4*

In the beginning, the CDV of the learner contains only one record regarding his difficulty level towards the first *Learning Object* LO_1. Each time he advances in the course, his CDV_{ij} is filled with more information and keeps growing until the learner reaches LO_n. At this point, he is considered a graduate (succeeder), and his stored CDV is tagged as "Success" (see the example presented in Figure 4.20). Otherwise, if the learner fails to reach the LO_n in time, his CDV_{ij} remains incomplete and is tagged as "Dropout" (see the example presented in Figure 4.19). In either case, this CDV is used afterward to predict newcomer learners whose CDVs are still incomplete (see Subsection 4.6.3)

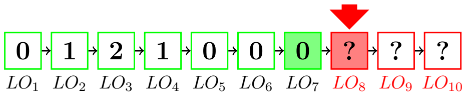


Figure 4.19 – *CDV of a dropper*

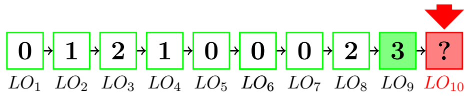


Figure 4.20 – *CDV of a succeeder*

4.6.2 Subsystem 2: The Difficulty Detection Subsystem (Same as Contribution 1)

This subsystem does not have many changes from the previous contribution (see 4.5.2);

4.6.3 Subsystem 3: The Difficulty Prediction Subsystem

The role of this subsystem is to predict learners who are most likely to drop out based on data stored in their *LearnerModels*. The "*Difficulty Detection Subsystem*" keeps updating the learner's difficulty history and stores it in his *LearnerModel* as a numeric vector called *CourseDifficultyVector* (CDV) used afterward by the "*Difficulty Prediction Subsystem*" to calculate the *CourseDropoutProbability* (CDP) and thus the *CoursePredictedStatus* (CPS).

The CDP is the probability that a learner may drop out. At the same time, the CPS represents the predicted status of the learner. It could be one of two values: "Drop" or "Success". To calculate the CDP_{ij} of the $Learner_i$ in the $Course_j$,the "*Difficulty Prediction Subsystem*" tries to find a match of his current CDV_{ij} in the *DataBase* (see Subsec. 4.6.3.2). To simplifiy the explaination, we will call searched CDV_{ij}, the *SearchedVector* (SV), and all found CDVs matching the SV, the *FoundVectors* (FVs).

The prediction process is performed in two stages :

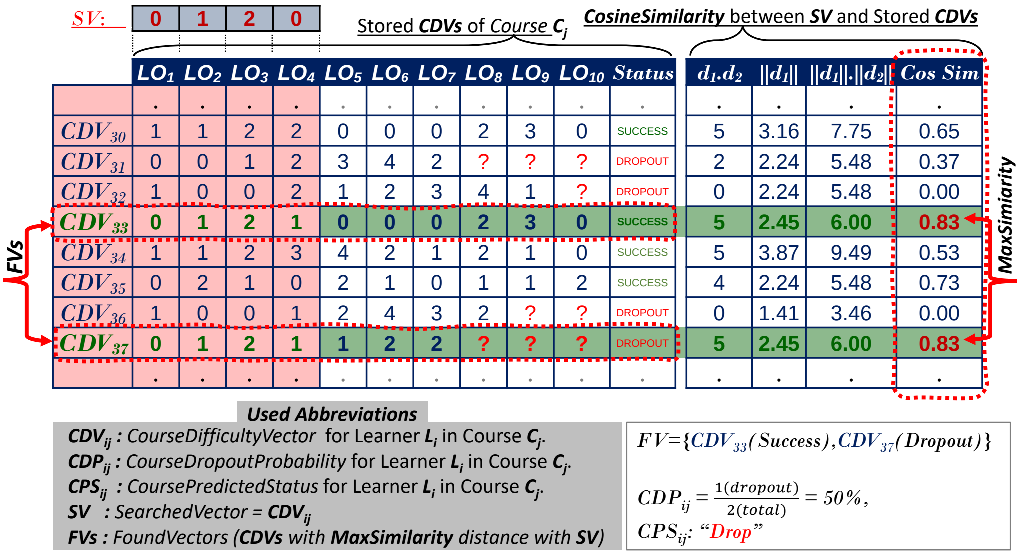
1. Find all similar vectors to SV (FVs).
2. Calculate CDP and CPS based on FVs stored data.

4.6.3.1 Found Vectors (FVs)

As shown in Figure 4.21, the "*Difficulty Prediction Subsystem*" starts by applying the ***CosineSimilarity*** function between SV and all present CDVs in the Learner Model DataBase to find the most similar vectors to SV, having the minximum similarity with SV.

The ***CosineSimilarity*** function is based on the Cosine Similarity technique (see Subsection 4.6.3.3). It determines how far two vectors are similar by measuring the cosine value of the angle between them. The smaller the angle between two vectors, the higher its cosine value, thus the similarity between them. Returned values are between "-1" and "1." A "-1" value means the vectors are opposite in direction, and there is no similarity between them. In contrast, a "1" value means that they are identical (Figure 4.22).

We called the highest returned cosine value the *MaximumSimilarityValue* (MaxSimilarity) used by the "*Difficulty Prediction Subsystem*" to identify all FVs. *FVs* are the set of all the CDV vectors having a *CosineSimiarity* value with SV equals



	LO_1	LO_2	LO_3	LO_4	LO_5	LO_6	LO_7	LO_8	LO_9	LO_{10}	Status	$d_1.d_2$	$\|\|d_1\|\|$	$\|\|d_1\|\|.\|\|d_2\|\|$	Cos Sim
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
CDV_{30}	1	1	2	2	0	0	0	2	3	0	SUCCESS	5	3.16	7.75	0.65
CDV_{31}	0	0	1	2	3	4	2	?	?	?	DROPOUT	2	2.24	5.48	0.37
CDV_{32}	1	0	0	2	1	2	3	4	1	?	DROPOUT	0	2.24	5.48	0.00
CDV_{33}	0	1	2	1	0	0	0	2	3	0	SUCCESS	5	2.45	6.00	0.83
CDV_{34}	1	1	2	3	4	2	1	2	1	0	SUCCESS	5	3.87	9.49	0.53
CDV_{35}	0	2	1	0	2	1	0	1	1	2	SUCCESS	4	2.24	5.48	0.73
CDV_{36}	1	0	0	1	2	4	3	2	?	?	DROPOUT	0	1.41	3.46	0.00
CDV_{37}	0	1	2	1	1	2	2	?	?	?	DROPOUT	5	2.45	6.00	0.83
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

Figure 4.21 – *Sample of stored CDVs (Course with 10 LOs)*

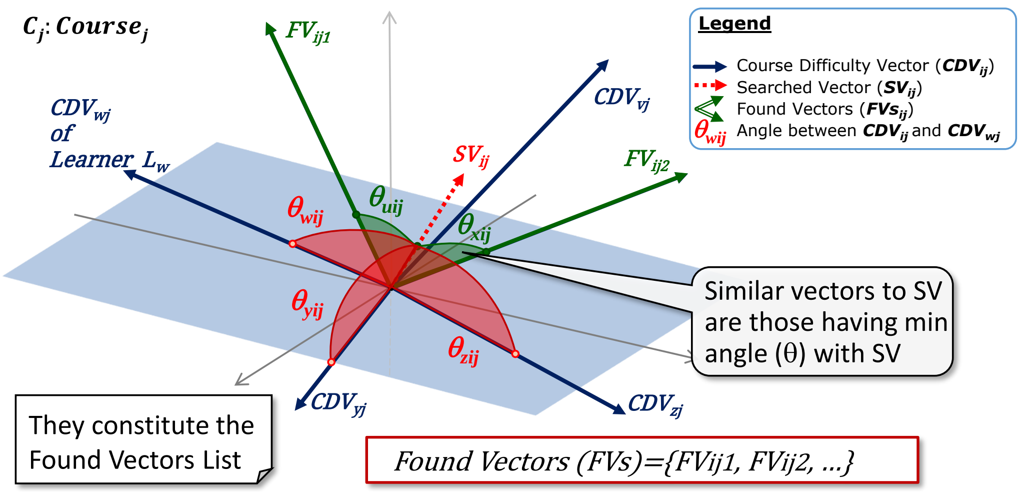


Figure 4.22 – *Using Cosine Similarity to find FVs Similar Vectors to SV*

to the *MaxSimilarity*; This means that FVs represent the set of vectors that have the minimum angle θ with SV.

4.6.3.2 Course Predicted Status (CPS)

Each one of the FVs has one of two states, "Dropout" or "Success" (Figure 4.21). The "*Difficulty Prediction Subsystem*" counts all the FVs presenting the state "Dropout," then divides this number by the total number of FVs to find the CDP_{ij} (Formula 4.7).

$$CDP_{ij} = \frac{\text{Number Of } FVs \text{ tagged as "Dropout"}}{\text{Total number of } FVs} \tag{4.7}$$

The CPS_{ij} is calculated conforming to Formula. 4.8.

$$CPS_{ij} = \begin{cases} \text{"Drop"}, & \text{if } (CDP_{ij} \geq 50\%) \\ \text{"Succeed"}, & \text{otherwise} \end{cases} \tag{4.8}$$

The whole prediction process is summarized in Algorithm 1.

Algorithme 1 : CPS_{ij} Calculation Algorithm (of the *Learner* L_i in the *Course* C_j)

Result : Calculate the *CoursePredictedStatus* (CPS_{ij}) of the *Learner* L_i in the *Course* C_j

$SV \leftarrow CDV_{ij}$; ; // *SearchedVector* SV = current CDV of the Learner L_i of Course C_j

begin

1 **Find <u>*MaxSimilarity*</u> value**;

$MaxSimilarity \leftarrow -1$; ; // Init *MaxSimilarity* with min value (-1)

$l \leftarrow 0$; ; // l is a counter to browse $StoredCDVs(C_j)$

for *(each* $CDV_{lj} \in StoredCDVs(C_j)$*)* **do**

  $CosSim \leftarrow CosineSimularity(SV, CDV_{lj})$;

  if $CosSim > MaxSimilarity$ **then**

    $MaxSimilarity \leftarrow CosSim$;

  end

end

2 **Find all <u>*FoundVectors (FVs)*</u> with <u>*similarity==MaxSimilarity*</u>**;

$NbFVs \leftarrow 0$; // Init the number of FVs with 0

$NbFVsTaggedDropout \leftarrow 0$; // Init the Number of FVs tagged as Dropout with 0

$l \leftarrow 0$; ; // l is a counter to browse $StoredCDVs(C_j)$

for *(each* $CDV_{lj} \in StoredCDVs(C_j)$*)* **do**

  $CosSim \leftarrow CosineSimularity(SV, CDV_{lj})$;

  if $CosSim = MaxSimilarity$ **then**

    if $FVs[Dropout] = Dropped$ **then**

      $NbFVsTaggedDropout \leftarrow NbFVsTaggedDropout + 1$;

    end

    $NbFVs \leftarrow NbFVs + 1$;

  end

end

3 **Calculate <u>CDP</u> and <u>CPS</u>**;

$CDP \leftarrow NbFVsTaggedDropout / NbFVs$; ; // $CDP \leftarrow \frac{NbFVsTaggedDropout}{NbFVs}$

if $CDP >= 0.5$ **then** $CPS \leftarrow$ "*Drop*";

else $CPS \leftarrow$ "*Succeed*";

end

4.6.3.3 The Cosine Similarity

We used the Cosine Similarity algorithm to find the most similar vectors to SV in the database. Also called cosine measure, this algorithm calculates the Similarity between two n-dimensional vectors. The Similarity is determined by computing the cosine of the angle between them. This algorithm is commonly used in text-mining problems. To explain more, let be two vectors A and B, the angle θ is obtained by the scalar product and the norm of the two vectors according to formula 4.9.

$$\cos\theta = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{4.9}$$

Where "·" represents the scalar product, and A means the magnitude of *"d."* As we know, values of $cos(A, B)$ are located in the range [-1,1], so "-1" means that A and B are opposites, and "0" means that those vectors are independent (orthogonal) while the value "1" means that those vectors are similar (col-linear of positive coefficient). In our case, we compute the Similarity between SV and each (CDV) in the database starting from the beginning (see equation 4.10). The vectors with the higher value of Similarity are the *FoundVectors*(FVs) set, used afterward to calculate the CDP and CPS

$$\cos\left(SV, CDV\right) = \frac{SV \cdot CDV}{||SV|| \cdot ||CDV||} \tag{4.10}$$

4.6.4 Subsystem 4: The Intervention Subsystem (Same as Contribution 1)

This subsystem has not undergone any significant changes in this contribution and is almost similar to the first contribution (see 4.5.3).

4.6.5 Intelligent Agents

As the approach has been updated in this second contribution to achieve prediction tasks, We added a new type of agent to carry out the learning difficulties prediction tasks. In total, five types of intelligent agents are used; they are distributed as follows:

A. **The Learner Model Update Subsystem:** uses one type of agent:
   - **PersonalAgent:** same as before (see Subsec. 4.5.4).

B. **The Difficulty Detection Subsystem:** rely on two types of agents:
   - **ContinuousControlAgent:** same as before (see Subsec. 4.5.4).
   - **DifficultyDetectionAgent:** same as before (see Subsec. 4.5.4).

C. **The Difficulty Prediction Subsystem:** uses one type of agent:
   - **DifficultyPredictionAgent:** This agent predicts the learner's final situation of whether he may graduate or drop out. By retrieving the learner's *CourseDifficultyVector* (CDV) and compare it to all CDVs stored in the database to find a set of similar vectors, this agent can compute his *CourseDropoutProbability* (CDP) and so his *CoursePredictedStatus* (CPS). If the CPS is predicted as "Drop out," it notifies the intervention agent. One instance per course is triggered on the course's first day and discarded on the last one.

D. **The Intervention Subsystem:** needs one type of agent:

- **InterventionAgent:** same as before (see Subsec. 4.5.4).

The following section describes the nature and numbers of the chosen agents and provides other information.

4.7 Structure, Tasks, Objectives, and Functions of each Agent in the Proposed MAS to Detect and Predict Learning Difficulties

The proposed approach comprises intelligent and autonomous subsystems collaborating to accomplish the system's main goals with minimum human intervention. However, the number of learners and courses is enormous compared to the number of subsystems, which could make the system more sequential-like and less responsive. The solution was to empower each subsystem with a set of intelligent agents to undertake its tasks for each learner and each course individually. As shown in Figure. 4.23, four types of agents are used.

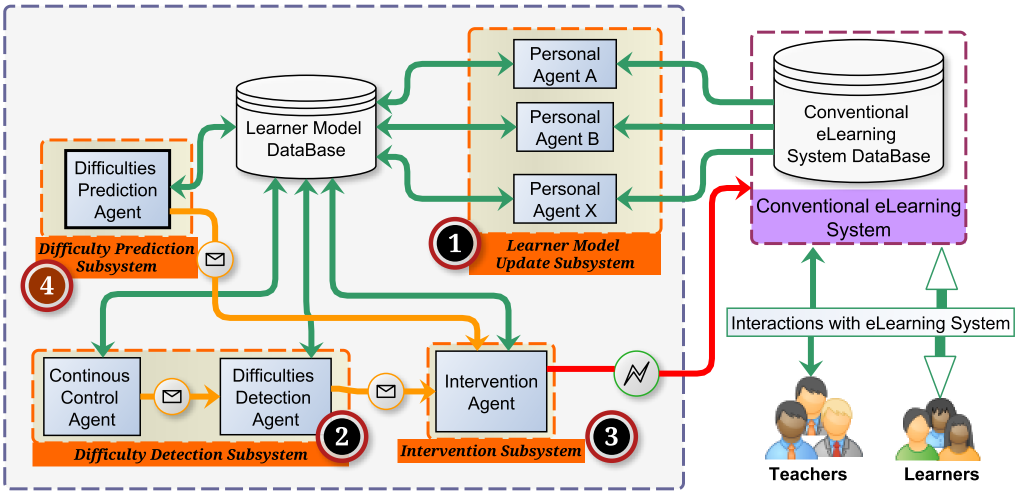


Figure 4.23 – *Proposed Multi-Agent System for Learning Difficulties Detection and Prediction in a given Course*

Table. 4.2 presents how the five agents are distributed among subsystems

Subsystem Name	**Agent Name**	**Type**	**Number**
Learner Model Update Subsystem	Personal Agent	Cognitive	1/Learner
Difficulty Detection Subsystem	Continuous Control Agent	Cognitive	1/Course
Difficulty Detection Subsystem	Difficulty Detection Agent	Cognitive	1/Course
Difficulty Prediction Subsystem	Difficulty Prediction Agent	Cognitive	1/Course
Intervention Subsystem	Intervention Agent	Cognitive	1/Course

Table 4.2 – *Intelligent Agents Distribution Among Proposed Subsystems.*

4.7.1 The Learner Model Update Subsystem:

This subsystem is responsible for gathering and filtering the learners' traces and building and updating the Learners' Models. This subsystem performs several tasks on each learner individually. It uses one type of agent called *PersonalAgent*.

1. *PersonalAgent*:

This agent creates and then keeps updating the contextual Model of the Learner. One instance of this agent is associated with each Learner, and each time the Learner access the System, his PersonalAgent is activated and put to work. This agent's features are gathered in Table. 4.3.

Name	**Personal Agent**
Tasks	- It creates new contextual model for newly enrolled *Learners*. - It updates the Learner contextual model towards all *LOs* in all *Courses*.
Interactions	- Conventional e-Learning System.
Objectives	- Collects and Filters the learners' traces to feed the Learner Model. - Creates and Updates the Learner Model.
Number	- One agent per learner
Triggering	- At the learner's first enrolment. - At each learner's login.
Funtions	**1. CreateModel:** This function creates the Learner Model during the first enrollment. **2. UpdateModel:** This function updates the Learner Model with each performed action of the learner
Database	- Learner Model Database

Table 4.3 – *Personal Agent Features*

4.7.2 The Difficulty Detection Subsystem:

This subsystem is responsible for the actual detection process, divided into two categories of tasks, time-related and performance-related tasks, each performed by a different type of agent. As a result this subsystem uses two types of agents, the *ContinuousControlAgent* and the *DifficultyDetectionAgent*.

2. *ContinuousControlAgent*:

The first set of tasks constitutes a set of daily-repeated tasks to check for all enrolled learners in one course. One agent called *ContinuousControlAgent* performs these tasks. One agent per course is assigned to check all the enrolled learners' access times. This agent performs all time-related tasks and checks the learners' persistence. This agent is triggered one time each day. This agent performs all time-related tasks to check checks the learners' persistence; for example, it can check the following:

- Delays related to LO's activities.

- Expiration of the period of a given test.
- A total absence of the learner

As soon as it detects an anomaly, this agent notifies the *DifficultyDetectionAgent* to check the learner's condition using other indicators to reassess the learner's "Difficulty Level".

This agent's features are gathered in Table. 4.4.

Name	**Continuous Control Agent**
Tasks	- Analyzes the different models of each learner and assesses whether or not the learner is behind learning at the same time for assessment. - It analyzes the Learners' Models to evaluate if a learner is falling behind and updates the Learner Model accordingly. - It notifies the *DifficultyDetectionAgent* if delays passed thresholds so it reassesses the *DifficultyLevels* using other indicators.
Interactions	- DifficultyDetectionAgent. - Conventional e-Learning System.
Objectives	- Follow-up of the learners attendance in academic activities . - Follow-up of essential dates (start and end of each activity)
Number	- One agent per learner
Triggering	- At the learner's first enrolment - Each day.
Funtions	**1. Count Learning Delay:** it checks for each learner's access times to all due LOs. **2. UpdateModel:** it updates the learner's model, used later by the Difficulty Detection Agent to evaluate the learner's indicators like his state. **3. NotifyDDAgent:** If the number of delay days is more significant than a certain threshold, it notifies the *Difficultydetectionagent*.
Database	- Learner Model Database

Table 4.4 – *Continuous Control Agent Features*

3. *DifficultyDetectionAgent*:

The second set of tasks is related to each learner's achievement and needs to be triggered for each learner's access to the "Evaluation Activities" or by a notification from the *ContinuousControlAgent*. These tasks are performed by an agent called *DifficultyDetectionAgent* per learner and responsible for assessing Primary (see Subsec. 4.5.2.1) and Secondary (see Subsec. 4.5.2.1) indicators.

This agent calculates the CourseDifficultyLevel of every Learner in real-time. In case of any detected difficulties, it notifies the intervention agent. One instance of this agent is associated with each course. This agent is triggered on the first day of the course and discarded on the last day. This agent's features are gathered in Table. 4.5.

Name	**Difficulty Detection Agent**
Tasks	- It calculates the *DifficultyLevel* for each LO for each Course - It analyzes each Learner Model to assess *DifficultyLevel* of that learner for each LO for each enrolled Course. - It updates the Learner Model accordingly. - It notifies the intervention agent to take the necessary action if any difficulties are detected.
Interactions	- ContinuousControlAgent. - InterventionAgent - Convetional e-Learning System.
Objectives	- Calculates each learner's learning difficulties levels for each enrolled course - Ensures that learners can pursue their learning without problems by identifying those who have problems, so interventions target them.
Number	- One agent per course
Triggering	- At the first enrolment of each learner in the course. - Notification from the *ContinuousControlAgent*.
Funtions	**1. CaclulateIndicators:** it calculates the *DifficultyLevel* for each learner in each LO, used later to find the course difficulty level for each learner. **2. UpdateModel:** it updates the learner's model with the learner's calculated status, indicators, and variables. **3. NotifyIntervAgent:** If a learner is identified as struggling, it sends a message to notify the Intervention Agent.
Database	- Learner Model Database

Table 4.5 – *Difficulty Detection Agent Features*

4.7.3 The Prediction Subsystem

As previously explained in Subsec. 4.6.3, this subsystem is the heart of the prediction processes. It analyzes the current difficulty histories of the current learners stored in his model and seeks to find similarities with previous learners' difficulties histories. It then applies the prediction algorithm to foresee the learners' future probable difficulties. In such cases, it notifies the Intervention subsystem to take the necessary measurements. This subsystem delegates a set of agents called *DifficultiesPredictionAgents* to perform his tasks.

4. The *DifficultiesPredictionAgent*

This agent predicts learning difficulties that a learner may face by predicting his final situation of whether he may graduate or drop out. By retrieving the learner's *CourseDifficultyVector* (CDV) and compare it to all CDVs stored in the Learner Model Database to find a set of similar vectors, this agent can compute his *CourseDropoutProbability* (CDP) and so his *CoursePredictedStatus* (CPS). If the CPS is predicted as "Drop out," it notifies the intervention agent. One instance per course is triggered on the course's first day and discarded on the last one. This agent is triggered two weeks after the course begins. This agent's features are gathered in Table. 4.6.

Name	Difficulty Prediction Agent
Tasks	- Analyzes each learner's *CourseDifficultyVector* (CDV) to ensure it has at least two records. - Compares each learner's CDV with previous learners' CDVs to find similarities. - Applies the prediction algorithm on the current learner's CDV and the most similar found. - In case of any predicted difficulties notifies the *InterventionAgent*.
Interactions	- InterventionAgent - Convetional e-Learning System.
Objectives	- Ensures early predictions of learners with learning difficulties to buy precious time to intervene and assist those learners.
Number	- One agent per course
Triggering	- Every day from the second week of the course start date.
Funtions	**1. FindSimilarities:** Analyzes each learner's CDV. If it contains at least two records, it starts seeking similar CDVs in the Learners Model Database. **2. DropoutPrediction:** Applies a prediction algorithm on the current learner CDV, and all found CDV to find a dropout probability and estimate a dropout status **3. UpdateModel:** it updates the learner's model, with the learner predicted probability and status. **4. NotifyIntervAgent:** If a learner is predicted as a potential dropout, it sends a message to notify the Intervention Agent.
Database	- Learner Model Database

Table 4.6 – *Difficulties Prediction Agent Features*

4.7.4 The Intervention Subsystem

This subsystem is responsible for taking necessary actions to assist learners tagged as struggling once it gets a notification from either the Detection or the Prediction subsystems. This subsystem needs one type of agent called the *InterventionAgent*.

Name	Intervention Agent
Tasks	- It takes the necessary actions to assist targeted learners - Sends messages to learners and teachers about severe detected *DifficultyLevel*.
Interactions	- DifficultyDetectionAgent. - Convetional e-Learning System.
Objectives	- Provides necessary help and assistance to struggling learners - Alert teachers about learners with difficulties.
Number	- One agent per course
Triggering	- Notification by the Difficulty Detection Agent.
Funtions	**1. SendMessageToLearner:** It sends a message to the learner about his performance. **2. SendMessageToTeacher:** Send a message to the teacher about the learner's performance.
Database	None

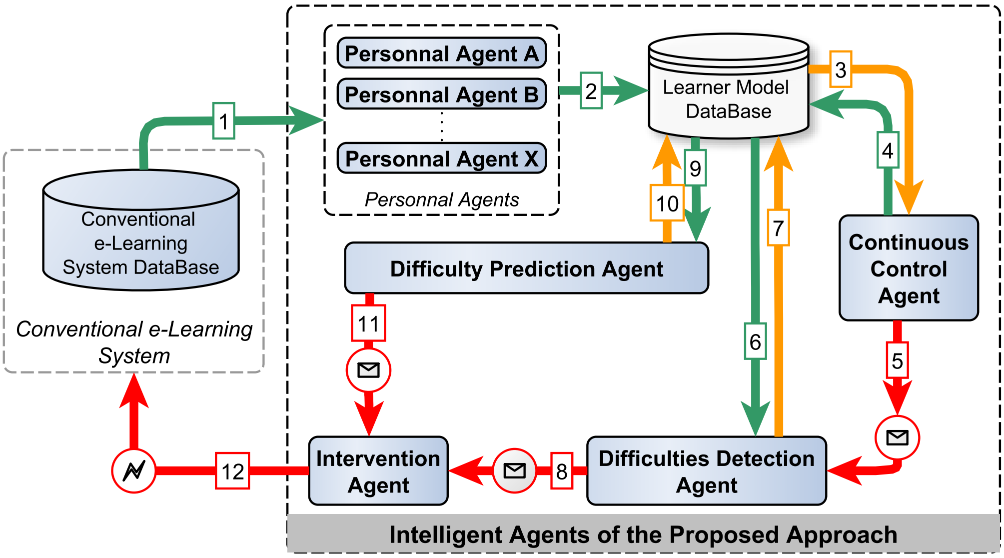
Table 4.7 – *Intervention Agent Features*

5. The *InterventionAgent*

The role of this agent is to take action to save detected or predicted in-distress learners. One instance of this agent is associated with each course. All this agent's features are gathered in Table. 4.7.

4.7.5 Scenario of Learning Difficulties Detection and Prediction Using the Proposed Multi-Agent System

Figure 4.24 illustrates the detection, prediction, and intervention scenario undertaken by the proposed system agents on a given *Learner A*. All taken actions are enumerated and explained. The figure also illustrates the complete architecture of the proposed intelligent system, including all interactions between the different agents' modules and the system.



Legend.

1 – Collect traces left by Learner A while interacting with the conventional e-learning system.
2 – Update –or creation if it does not exist– of the Model of Learner A.
3 – daily Retrieval of all the activities access dates of Learner A.
4 – Update of all the time-related indicators in Learner A Model.
5 – Warn the "Difficulty Detection Agent" of any detected time-related problem with Learner A.
6 – Read the computed learner's difficulty indicators and evaluation of the learner's level of difficulty.
7 – Update of the current difficulty level of Learner A in his Model.
8 – Send notification to the Intervention Agent If difficulties are detected.
9 – Read the current learner's CDV from his Model.
10 – Update the Predicted status in the Model of Learner A.
11 – Send a notification to the Intervention Agent if the predicted status is "Dropout."
12 – Provide the conventional e-learning system with the suggested intervention actions to be taken.

Figure 4.24 – *Scenario of interactions between the proposed Intelligent Agents*

4.8 Conclusion

In this chapter, we proposed a new agent-based approach used as an Early Warning System (EWS) to detect and predict the behavior of learners facing difficulties. We proceeded with two successive and complementary contributions to come to a final approach that can achieve all previously mentioned objectives and answer these research questions.

In the first contribution, we focused on indicators of difficulties and the modeling of learners in such a way as to identify those learners in difficulty and also intervene autonomously to assist them. The proposed approach calculates the level of difficulty for each course and each Learning Object for each learner (enrolled in that course) based on a set of primary and secondary indicators used to deduce his situation concerning this course. We also proposed a MAS capable of implementing this approach.

The second contribution is the successor of the first one, where we proposed a second contribution based on the first one and were able to predict the learning difficulties of learners and those who will encounter them. For this purpose, the proposed approach calculates a difficulty vector for each learner enrolled in each course. This difficulty vector represents the history of difficulties of this learner in this course. Furthermore, the proposed approach computes the distance between the learner's difficulty vector and those of his peers to search for similarities. The results are then used to predict the situation of this learner. We also updated our proposed MAS to support the new proposed contribution.

The proposed approaches have been implemented in two consecutive prototypes. The second prototype is a senior update of the first, allowing the detection, prediction, and assistance of struggling learners in human learning systems.

The following chapter will present these prototypes and the results of different experiments conducted with real students to validate the theoretical proposals presented in this chapter.

Chapter 5

Prototypes implementation, Experiments Results, and Discussions

Contents

5.1 Introduction . . . 112
5.2 Building Prototypes . . . 112
 5.2.1 *Development Tools to Build the Prototypes* . . . 112
5.3 LearnDiP+ Functionalities . . . 115
 5.3.1 *LearnDiP+ for the Students* . . . 115
 5.3.2 *LearnDiP+ For the Teachers* . . . 115
5.4 Accessing and Using LearnDiP+ . . . 117
 5.4.1 *Teacher Space* . . . 120
 5.4.2 *Admin Space* . . . 124
 5.4.3 *Student Space* . . . 125
5.5 Experimentations, Results and Discussion . . . 130
 5.5.1 *Experimentation I: Testing the effectiveness of the prototype to detect at-risk learners and intervene to enhance their cognitive level and diminish dropouts number* . . . 131
 5.5.2 *Experimentation II: Testing the effectiveness of the prototype to predict struggling learner dropouts* . . . 134
5.6 Conclusion . . . 137

5.1 Introduction

The previous chapter presented the proposed approach to deal with the problem of detecting and predicting learners with learning difficulties. It also presented the different proposed theoretical ideas and suggestions alongside their different components and necessary steps that could be used to implement them. This chapter introduces developed computer tools used to build two prototypes to enforce and validate the different proposed ideas. These two prototypes used a set of new technologies that were bound together to form a whole learning system to attain primary fixed objectives alongside the ordinary pedagogical tasks of any learning system.

The main objective of building such prototypes is to be able to conduct various experiments to support and validate our proposed approaches. Experiments were conducted with real students from the University of 8 Mai 1945 Guelma to explore, observe and validate the different aspects of our proposals. This chapter also presents the nature of conducted experiments, their participants, used methods and statistical tests, and obtained results.

5.2 Building Prototypes

Two prototypes related to the proposed contributions are built and used to conduct experiments to analyze and validate the proposed approaches. The first built prototype, named "LearnDiP" (Learning Difficulties Prevention), was developed within the Laboratory of Sciences and Technologies of Information and Communication (LabSTIC) of the University of 8 May 1945 Guelma. It is based on the first contribution that aims to detect and identify students with learning difficulties based on their gathered and analyzed traces.

The second prototype is the follower of the first one. It is a senior update of the first renamed "LearnDiP+[1]" insinuating that it is a continuance of the first one. 3 The significant difference between the two prototypes is that LearnDiP+ integrates a predictive model granting a new huge feature to allow the prediction of students in difficulty in addition to their detection. As "LearnDiP+" is an update and a continuation of "LearnDiP", all the features of "LearnDiP" are present in "LearnDiP+". In the following, we present only the final prototype "LearnDiP+" implemented after completing the second contribution.

5.2.1 Development Tools to Build the Prototypes

Our prototypes were developed as dynamic websites, mainly using the PHP programming language, HTML, JavaScript, and CSS. It was deployed through an Apache HTTP server and the MySQL database engine. We also used "BootStrap" and "JQuery" libraries to improve our system readability and make the website more responsive, especially for smartphones and tablets. We also used the "FontAwesome" font library to have a wide range of icons to communicate more visually with students.

[1] LearnDiP+ *https://tinyurl.com/learndipplus* (Accessed: 19, May 2022)

In the following, we explain all the technologies mentioned earlier:

5.2.1.1 The Apache HTTP Server

As determined by the Apache Software Foundation[2], the Apache HTTP Server Project is an effort to build and maintain an open-source HTTP server for modern operating systems such as UNIX and Windows. This project aims to provide a secure, efficient, and flexible HTTP server that complies with current HTTP standards.

5.2.1.2 The PHP Programming Language

As specified by The PHP Group[3], PHP is a recursive acronym for "Hypertext Preprocessor" it is "*a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.*"

5.2.1.3 HTML, CSS and JavaScript

HTML, CSS, and JavaScript are the three main languages used to build websites in web development.

HTML stands for" Hyper-Text Markup Language". It is defined by W3Schools[4] as the standard markup language for creating Web pages. HTML describes the structure of a Web page and consists of a series of elements that tell the browser how to display the content and label pieces of content.

All the styles in a webpage are moved to a new type of file separated from HTML called CSS to separate presentation from content in a website. The Cascading Style Sheets, shortly CSS, is "*is a style sheet language used to describe a document's presentation in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript*" As mentioned by wikipedia[5]. As specified by W3Schools [6] CSS is the used language to style an HTML document and describes how HTML elements should be displayed. It uses a different language from HTML. CSS allows consistent styling of elements across all pages on a website so that all headings, lists, and paragraphs look and act the same on every site page.

Alongside HTML and CSS, JavaScript, shortened as JS, is one of the core technologies programming languages on the World Wide Web. Most websites use JavaScript on the client side, usually including third-party libraries. As reported by Wikipedia [7],

---

[2] Apache HTTP Server *https://commons.apache.org/* (Accessed: 18, May 2022)

[3] PHP. *https://www.php.net/manual/en/intro-whatis.php* (Accessed: 18, May 2022)

[4] HTML. in *W3Shools*. *https://www.w3schools.com/html/html_intro.asp* (Accessed: 20, May 2022)

[5] CSS. in *Wikipedia*. *https://en.wikipedia.org/wiki/CSS* (Accessed: 20, May 2022)

[6] CSS. in *W3Shools*. *https://www.w3schools.com/css/default.asp* (Accessed: 20, May 2022)

[7] JavaScript. in *Wikipedia*. *https://en.wikipedia.org/wiki/JavaScript* (Accessed: 20, May 2022)

5.2.1.4 BootStrap

As maintained by Wikipedia[8], Bootstrap is a free and open-source CSS framework for front-end web development that is responsive and mobile-first. It includes design templates for typography, forms, buttons, navigation, and other interface elements in HTML, CSS, and (optionally) JavaScript. It can be accessed through: *https://getbootstrap.com/*

5.2.1.5 JQuery

As stated in Wikipedia[9], jQuery is a JavaScript library that makes traversing and manipulating the HTML DOM tree, event handling, CSS animation, and Ajax, easier. It is open-source software that is free to use under the MIT License. It can be accessed through: *https://jquery.com/*.

5.2.1.6 FontAwsome

As described in Wikipedia [10] Font Awesome is a font and icon library based on CSS and used to add icons to websites. It is accessible through: *https://fontawesome.com/*.

5.2.1.7 MySQL Relational Database Management System

As mentioned by Wikipedia[11], MySQL[12], is a Relational Database Management System that is free to use (RDBMS).

A relational database organizes data into one or more tables where data can be related to each other, allowing the data to be structured. SQL is a programming language that allows programmers to create, change, and extract data from relational databases and control user access. An RDBMS, like MySQL, in addition to relational databases and SQL, works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access, and makes database integrity testing and backup creation easier.

[8] Bootstrap (front-end framework). *Wikipedia* *https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)* (Accessed: 20, May 2022)

[9] JQuery. in *Wikipedia* *https://en.wikipedia.org/wiki/JQuery* (Accessed: 20, May 2022)

[10] FontAwesome. in *Wikipedia* *https://en.wikipedia.org/wiki/Font_Awesome* (Accessed: 20, May 2022)

[11] MySQL. in *Wikipedia*. *https://en.wikipedia.org/wiki/MySQL* (Accessed: 18, May 2022)

[12] What is MySQL? *https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html* (Accessed: 18, May 2022)

5.3 LearnDiP+ Functionalities

LearnDiP+ is a system that deals with three main actors, offering a set of functionalities for each. Each of these actors has a proper space to carry on his tasks. These spaces are the "Student Space," the "Teacher Space" and the "Admin Space."

5.3.1 LearnDiP+ for the Students

In addition to basic learning functionalities offered by any conventional learning environment, LearnDiP+ allows the student to be aware of his progress and current difficulty condition. In each step, the system informs the student of his finished and waiting tasks. The student can check his difficulty status using a dedicated set of pages that displays his progress, difficulty levels, and scores for all his courses, a specific course, or for each Learning Object separately. As shown in Figure. 5.1, the student can see his progress and difficulty level in the course, and all its Learning Objects (called chapters in the system to be more understandable for teachers and students)

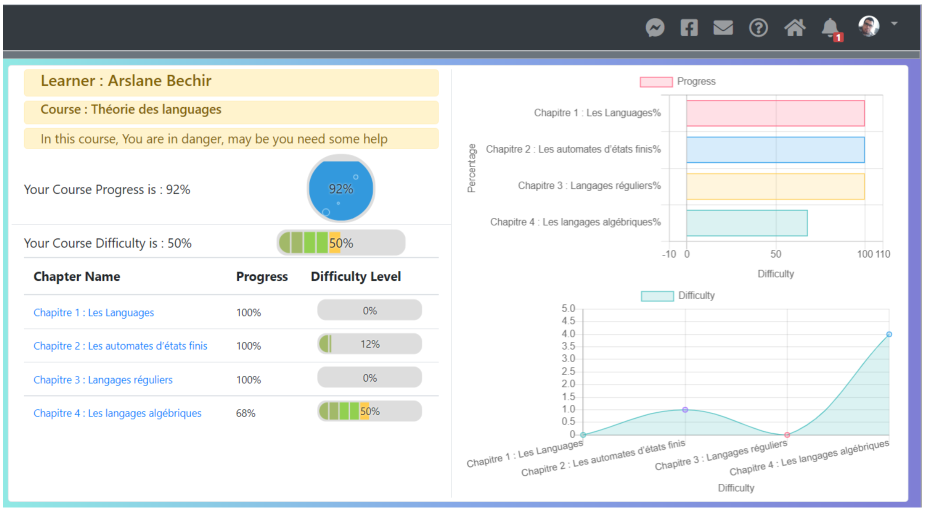


Figure 5.1 – *Student Progress and difficulty Levels of a given Course*

When the student chooses a particular Learning Object, he can see his progress and difficulty level in that Learning Object, and he can also see all his attempts' scores in that Learning Object's evaluation activity (Figure. 5.2).

5.3.2 LearnDiP+ For the Teachers

For the teacher, LearnDiP+ offers two major functionalities:

- **Learning Difficulties Detection** LearnDiP+ offers teachers real-time follow-up tasks to detect any learning difficulties.
- **Learning Difficulties Prediction** LearnDiP+ offers teachers predictive estimating values of eventual learning difficulties.

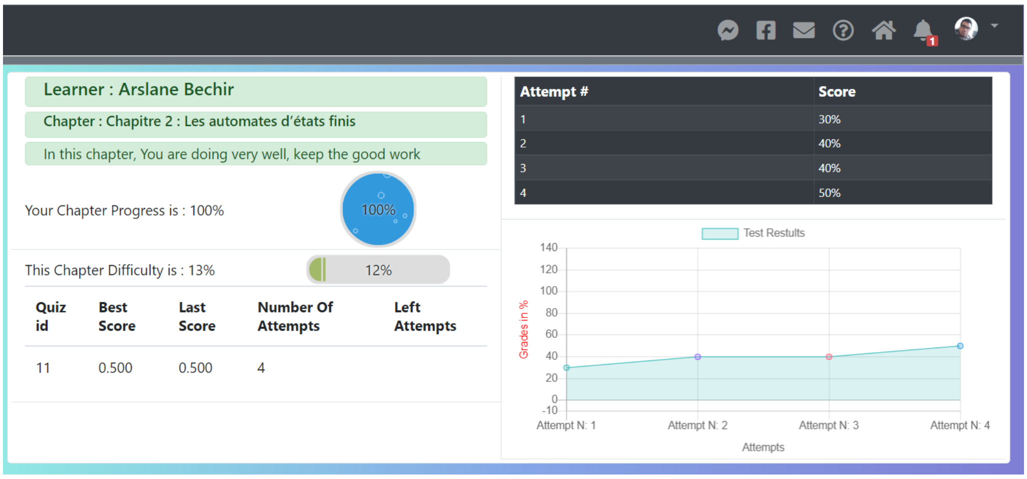


Figure 5.2 – *Student Progress and difficulty Levels of a given Learning Object*

5.3.2.1 Learning Difficulties Detection

The teacher can follow his students closely and see their achievements, progresses, and difficulty levels. For each course, the teacher has a broad overview of his course's enrolled students and has a set of values and actions as shown in (Figure. 5.3).

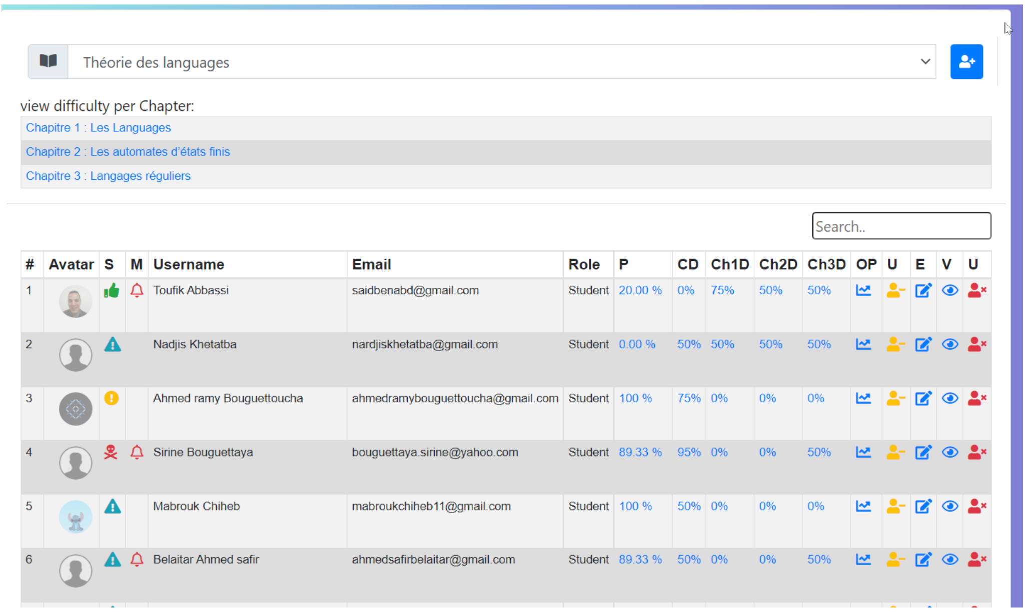


Figure 5.3 – *Student Progress and difficulty Levels of a given Learning Object*

All the values and actions are explained in Table. 5.1

5.3.2.2 Learning Difficulties Prediction

LearnDiP+ offers teachers the possibility to have a predictive overview of each course's enrolled students. Same as the detection part, the teacher has a complete list of students enrolled in his course with some displayed values and allowed actions as shown in (Figure. 5.4).

Abbreviation	Full Name	Functionnality
# & Avatar	Number & Profile picture	The number of the Student in the list and his picture
S	Difficulty Status	Icon representing the student difficulty status, using one of four icons with four colors
M	Monitored	For the experiment, students were divided into two groups Test and Control. The Test group students are Monitored and benefit from the system's features while the others do not.
Username & Email	Username & Email	Student username and Email
Role	Role	Other roles like Teacher or Admin
P	Course Progress	Student's progress in the course (in percentage)
CD	Course Difficulty Level	Student's Difficulty Level (in percentage) used to display his Difficulty Status
Ch1D & Ch2D & Ch3D	Difficulty Level for chapters 1,2,3	Student's Difficulty Level (in percentage) in the corresponding Learning Object (Chapter)
OP	Overall Progress	Student's Overall progress in all his enrolled courses (in percentage)
U	Unroll from the course	Button to unroll the student from the course but keep his history
E & v	Edit & View Profile	Edit and View the student profile
UD	Unroll and Delete progress	Button to unroll the student from the course and erase his history

Table 5.1 – *List of students enrolled in a given Course with all the real-time detection functionalities*

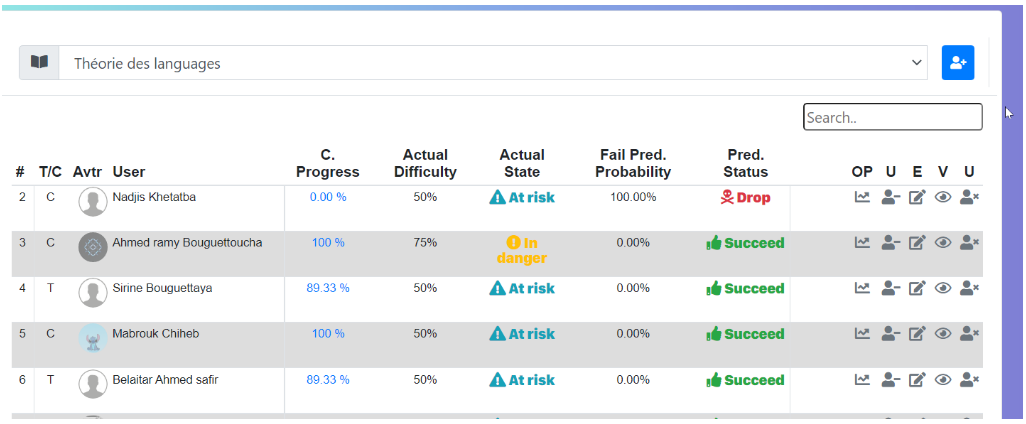


Figure 5.4 – *Student Predictions of a given Course*

Values and actions offered within this page are presented in Table. 5.2

5.4 Accessing and Using LearnDiP+

LearnDiP+ is accessible by following the link: *https://tinyurl.com/learndipplus*. The welcome interface will appear with an embedded YouTube video showing how to subscribe to the system. (see Figure. 5.5). The subscriber can consult the "Help Page", accessible from the "?" link, or contact the administrators directly through email or

Abbreviation	Full Name	Functionnality
#	Number	The number of the Student in the list
T/C	Test/Control	For the sake of the experiment, students were divided into two groups, Test and Control, where the Test group students benefited from the system's features while the others did not.
Avtr & User	Profile picture & Username	Student's Picture and username
C. Progress	Course Progress	Student's Progress in the course (in percentage)
Actual Difficulty	Course Difficulty Level	Student's Current Difficulty Level (in percentage) used to display his detected difficulty status
Actual State	Difficulty Status	Icon representing the student difficulty status, using one of four icons with four colors
Fail Pred. Probability	Failure Prediction Probability	Course Probability for this student to drop out (in percentage) used to display hid-predicted status
Pred. Status	Predicted Status	Course Predicted Status, whether he would succeed or drop
OP	Overall Progress	Student's Overall progress in all his enrolled courses (in percentage)
U	Unroll from the course	Button to unroll the student from the course but keep his history
E & v	Edit & View Profile	Edit and View the student profile
UD	Unroll and Delete progress	Button to unroll the student from the course and erase his history

Table 5.2 – *List of students enrolled in a given Course with all the system prediction functionalities*

Facebook in case of any problem. From the "Home Page" the visitor can follow multiple links to the other pages like the "Sign In" page to log in or the "Sign Up" page to subscribe if not already subscribed, or The "Reset Password" forgotten password to reset.

Students and teachers must register as system users before using the system. Each of them has to fill in a registration form with personal information, pick a username and a profile picture, and most importantly, choose the role "Student" or "Teacher."

The subscribers receive an email with an activation link that embeds a single-use expirable verification code

In case the user forgets the username or the password, he could always reset his password using the "Reset Password" page

After logging in, the user is automatically directed to his space, whether he is a teacher, an admin, or a student. The system has two interfaces: the FrontEnd for the students and the BackEnd for the teachers and admins.

1. **The Teacher Space:** The teacher space (Figure. 5.6) is the BackEnd of the website where teachers can manage courses, learning objects, and evaluation tests. Teachers can edit only their created materials or that assigned to them by administrators. They can also enroll or unenroll students in their assigned courses, access their profiles and information, and follow their progress and difficulties. Because the system allows a teacher to be a student in other courses, the teacher can switch to "Student Space" to follow up on the courses he is

Figure 5.5 – *LearnDiP+ Home Page*

enrolled in "as a student" or preview the material he created for his students (see Subsec. 5.4.1).

2. **The Admin Space:** The admin space (Figure. 5.7) is similar to the teacher space (Figure. 5.6), in addition to the "Administration" pane, where an admin can manage the system users and roles. Another difference is that the teacher can only control his content and his students; however, the admin can see and manage the content of all teachers and students. The account assigned permissions are visible in the "Login Infos." pane, whether it has Admin, Teacher, or Student rights. An account can have multiple permissions (Figure. 5.7)

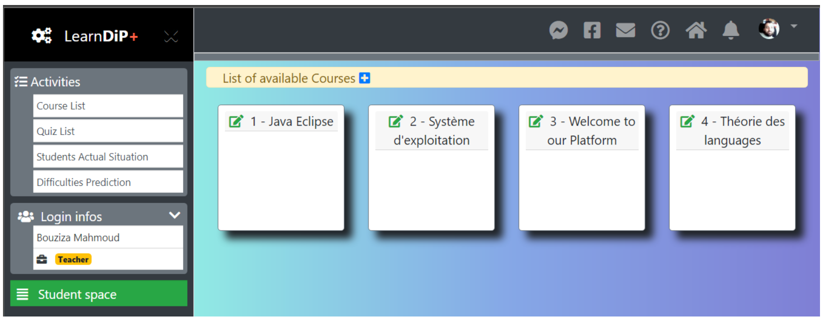


Figure 5.6 – *Teacher Space Welcome Page*

3. **The Student Space:** it is the FrontEnd of the website oriented to the students. It contains all the learning and evaluation materials. The student can see his progress, scheduled activities and evaluation tests, and most importantly, his performance, whether he is doing well or facing difficulties (see Subsec. 5.4.3).

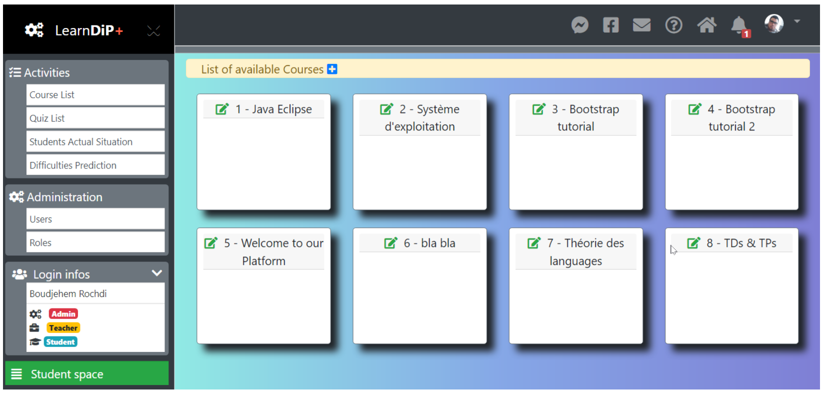


Figure 5.7 – *Admin Space Page*

5.4.1 Teacher Space

The teacher's space allows a teacher to achieve these major tasks:

1. Pedagogical Content Management
2. Enrolling Students
3. Learning Difficulties Identification
4. Learning Difficulties Prediction

5.4.1.1 Pedagogical content Management

The teacher is responsible for creating and editing the pedagogical content as the domain expert. The pedagogical content could be any learning or evaluation content:

a. The Learning Content: The learning content is grouped by courses that comprise Learning Objects; each comprises a set of Learning activities and Evaluation Activities. Through the "Course List" menu (Figure. 5.8), the teacher could create, edit or delete courses (Figure. 5.9), Learning Objects and Learning Activities He is also responsible for setting the starting and ending dates for each course and learning object, among other settings.

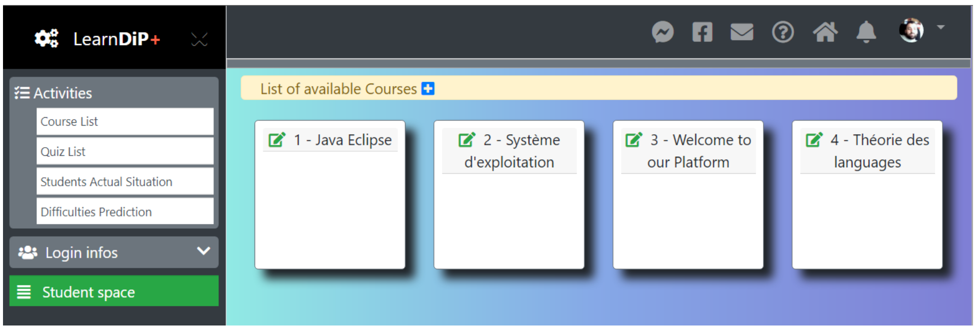


Figure 5.8 – *Teacher Editable Learning Content*

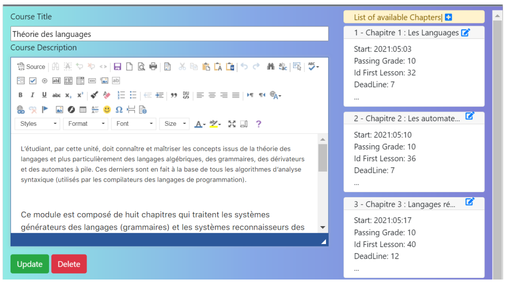


Figure 5.9 – *Teacher Courses Edition*

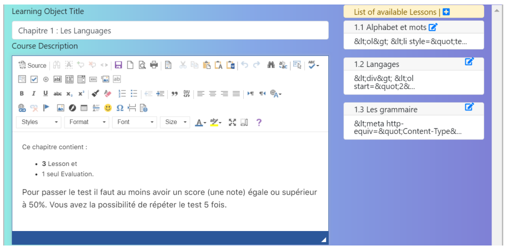


Figure 5.10 – *Teacher Learning Objects Edition*

b. The Students' Evaluation content: The primary evaluation means in our system is a set of quizzes that comprises Multiple Choice Questions (MCQ). Each quiz is composed of a set of questions.

Through the "Quiz List" menu, the teacher could create, edit or delete Quizzes (Figure.5.11), and Questions (Figure.5.12). He is also responsible for setting the right answer for each question.

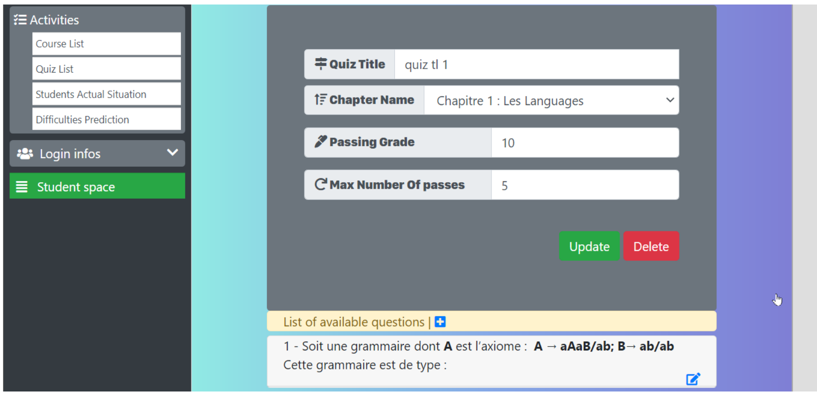


Figure 5.11 – *Teacher Quizzes Edition*

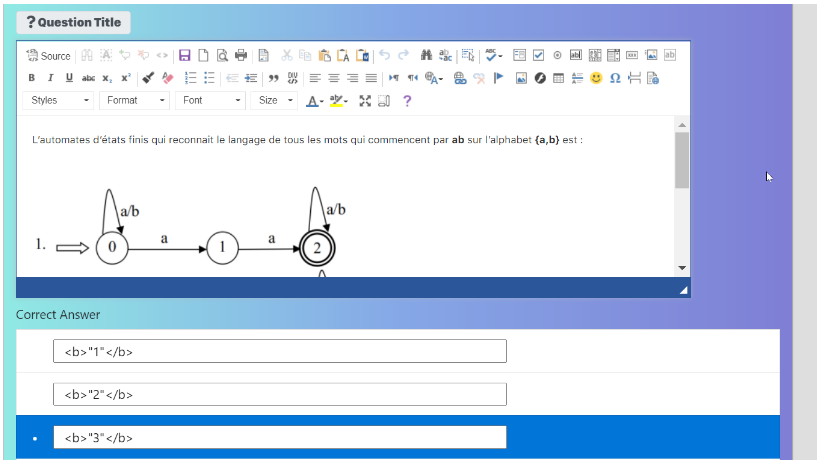


Figure 5.12 – *Teacher Questions Edition*

5.4.1.2 Enrolling students

To enroll a student, the teacher can access either the "Students Actual Situation" or "Difficulties Prediction" from the side menu. After choosing a course from the droplist, all students enrolled in this course are listed with all their related information and their condition, whether it is current (Figure. 5.13) or future (Figure. 5.14).

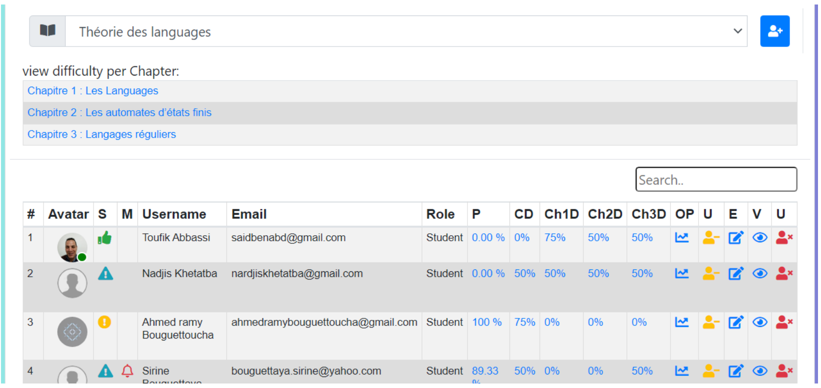


Figure 5.13 – *Detected Learning Difficulties in the course listed for each student*

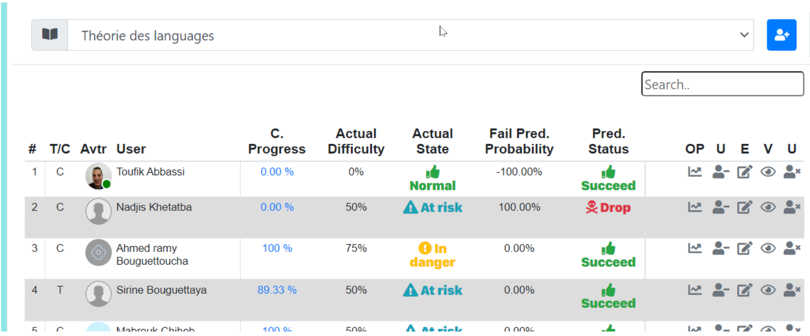


Figure 5.14 – *Predicted Learning Difficulties in the course listed for each student*

From this page, the teacher can enroll a student by clicking on the "Register" button and following the steps shown in (Figure. 5.15). After enrolling the student, he is enlisted on the page, and all his information regarding the course is displayed.

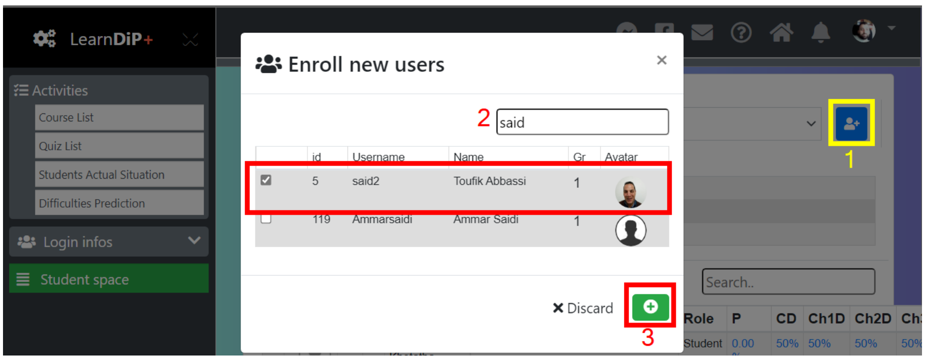


Figure 5.15 – *The Teacher enrolling a Student*

5.4.1.3 Learning Difficulties Identification

This page is the main page to manage the students per course. All students enrolled in this course are listed on this page and can be filtered through the "Search" box to display only some specific students (Figure. 5.16). The teacher (or Admin) can choose a specific chapter to follow up closely (Figure. 5.17). This page displays all students' general information, including their ids, full names, usernames, connection statuses, and profile pages.

Four color-coded icons are used to display the student's condition:

1.  : Very Good.
2.  : Good.
3.  : At-Risk.
4.  : In-Danger.

The teacher (or admin) can see the student's course progress and difficulty level alongside his difficulty level in each chapter. by clicking on these values the student's progress and difficulties will be displayed towards a specific course(Figure. 5.16) or a chapter(Figure. 5.17). The teacher also can enroll any user and reset all his progress in the course or just enroll him and keep his progress.

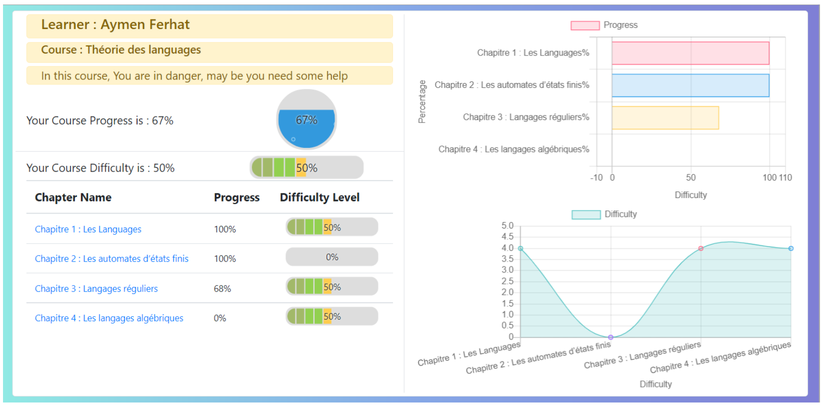


Figure 5.16 – *Students' Progress and Difficulties in the Course and Chapters*

5.4.1.4 Learning Difficulties Prediction

From this page, the teacher can see the system predictions for all enrolled courses and whether they will succeed or drop out. The actual course difficulty is also displayed, where the teacher could make a quick assessment of future endangered students. Figure. 5.14

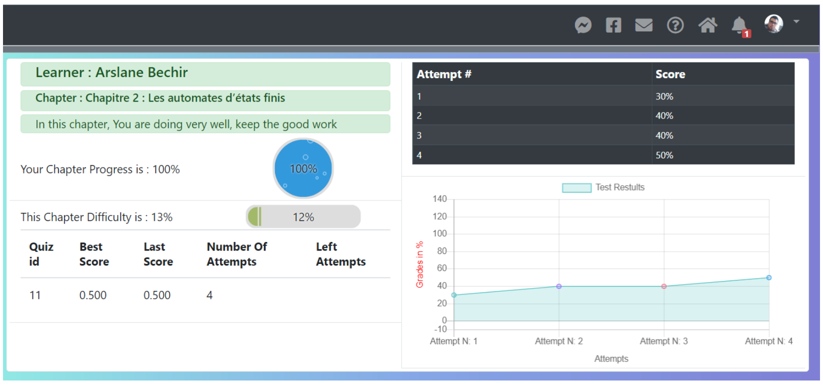


Figure 5.17 – *Students' Progress and Difficulties in a Selected Chapter*

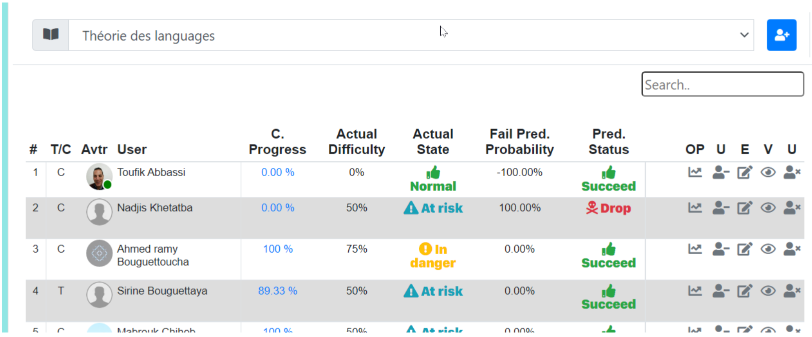


Figure 5.18 – *Predicted Learning Difficulties in the course listed for each student*

5.4.1.5 Chat with Students

The teacher can talk directly to online Students and teachers assigned to that course and is visible in the student and teacher space (Figure. 5.19).

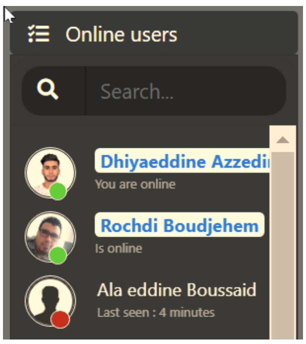


Figure 5.19 – *Online Students and Teachers*

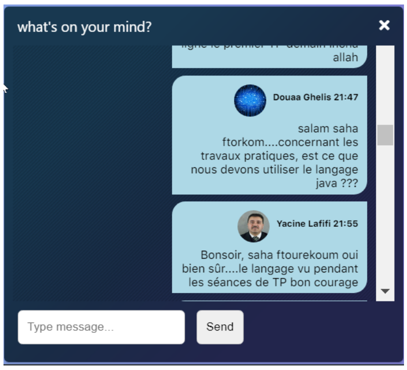


Figure 5.20 – *Chat between Teachers and Students*

The chat button is always visible, and when triggered, the chat window is open (Figure. 5.20).

5.4.2 Admin Space

An admin could perform all the teacher's tasks in addition to the system users management (Figure. 5.21) and roles management (Figure. 5.22).

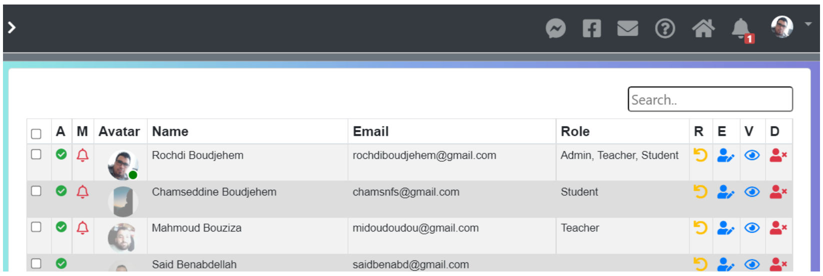


Figure 5.21 – *Adming Users Management*

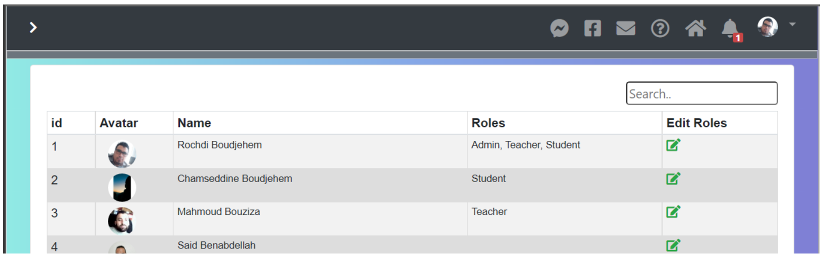


Figure 5.22 – *Adming Roles Management*

The admin can view, edit or delete any user in the system or simply reset all his settings as if he is newly subscribed. The admin can also assign permission to see or edit any course as a teacher or a student, and he can appoint other admins.

5.4.3 Student Space

After a successful subscription, the student should wait for his teachers to enroll him in their courses; until that, his space displays a notification that he is not registered in his course. After the teacher enrolls the student in his course, the course becomes visible to the student (Figure 5.23)

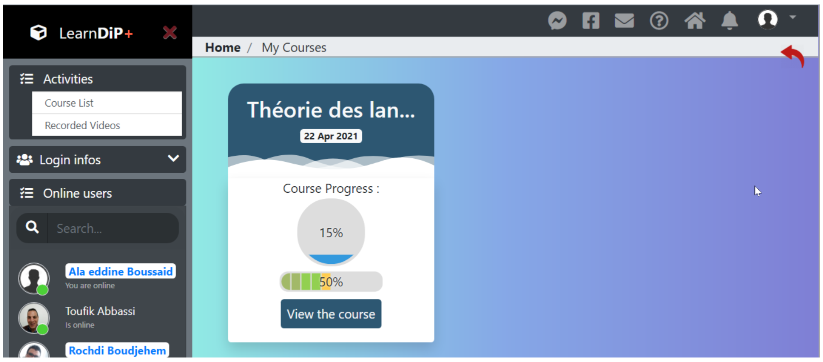


Figure 5.23 – *Student has Access to the Enrolled Course*

In his space, a student could perform these operations:

- Profile information View and Edit.
- Learning Content Browsing.
- Evaluations Passing.
- Self Aware Condition Check.
- Chat with teachers and peers.

5.4.3.1 Profile information View and Edit

The student can at any moment view his profile information or edit them by changing his name, phone number, password, or picture.

5.4.3.2 Learning Content Browsing

The student's list of enrolled courses is on the welcome page in the student space. He can access all his enrolled courses by clicking on the wanted course (Figure. 5.24). The course information is displayed with the course Learning Objects (chapters) and

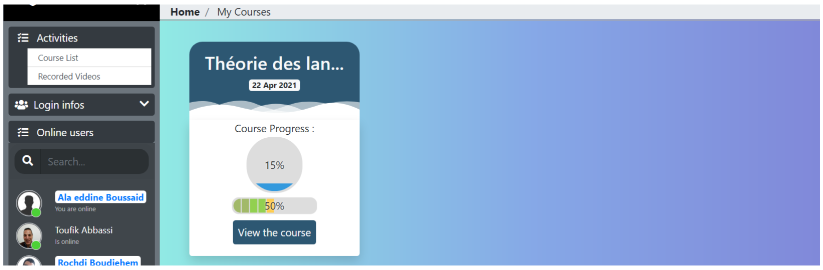


Figure 5.24 – *Student Course List*

evaluations (Figure. 5.25).

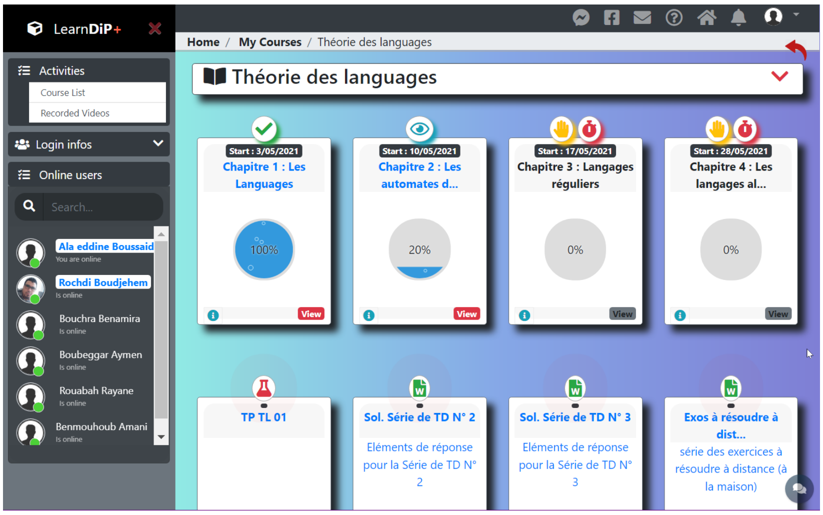


Figure 5.25 – *Course Content*

Not all the content is available to the student. Some content is timed, like the delayed chapters that do not start until specific dates. In contrast, other content relates to the student's progress and prerequisite achievements (Figure. 5.26).

We used different icons, colors, and tooltip information to guide the student. The chapter comprises a set of lessons accessible by clicking on the desired lesson (Figure. 5.27). We choose to implement only "HTML" non-downloadable lessons to ensure the learning phase is conducted within the system and be able to monitor the student.

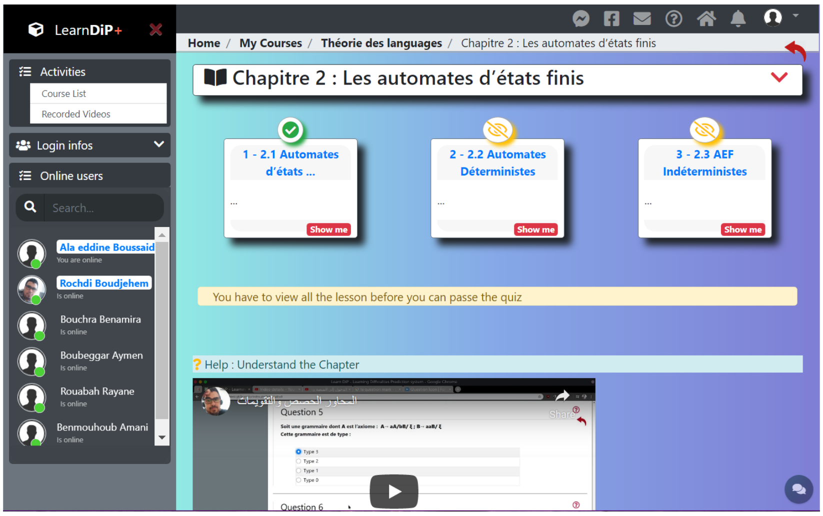


Figure 5.26 – *Chapters Content*

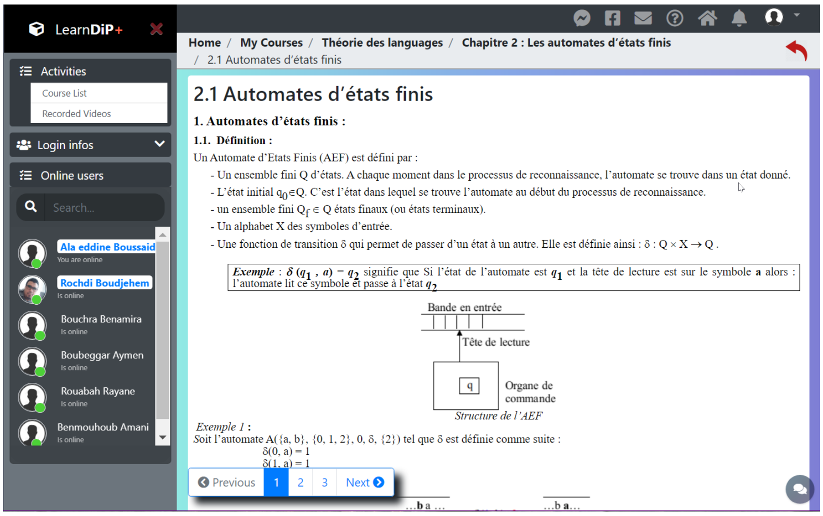


Figure 5.27 – *Lessons Content*

5.4.3.3 Evaluation Activities

The student should access all the available learning material before he can access the evaluation through the chapter (Figure.5.28) or the lesson (Figure.5.29).

The evaluations in our system are in the form of Multiple Choice Questions (MCQ), where the student can choose only one answer (Figure.5.30). The quiz is not limited by time or number. However, each quiz has a maximum attempt number to help assess the student's difficulty level. The scores and numbers of attempts are displayed on the chapter's page (Figure.5.31).

The evaluation result is displayed, and whether the student has succeeded (Figure. 5.32) in the evaluation test or has failed (Figure. 5.33), correspondent options are displayed to access the next chapter in the case of success or to repeat the evaluation or the chapter in the other case.

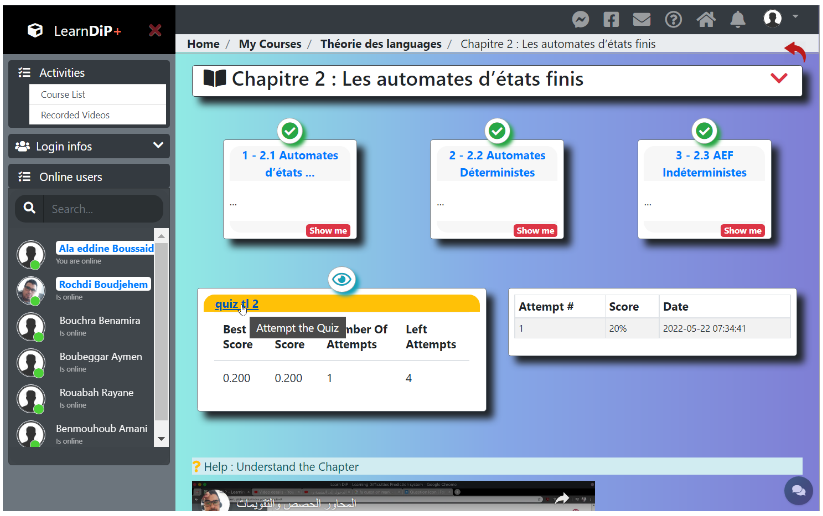


Figure 5.28 – *Evaluation Access through the Chapter*

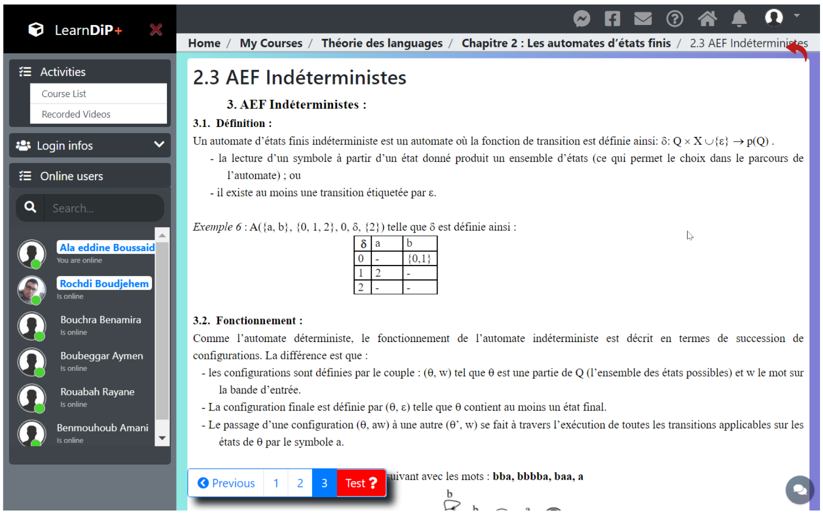


Figure 5.29 – *Evaluation Access through the Lesson*

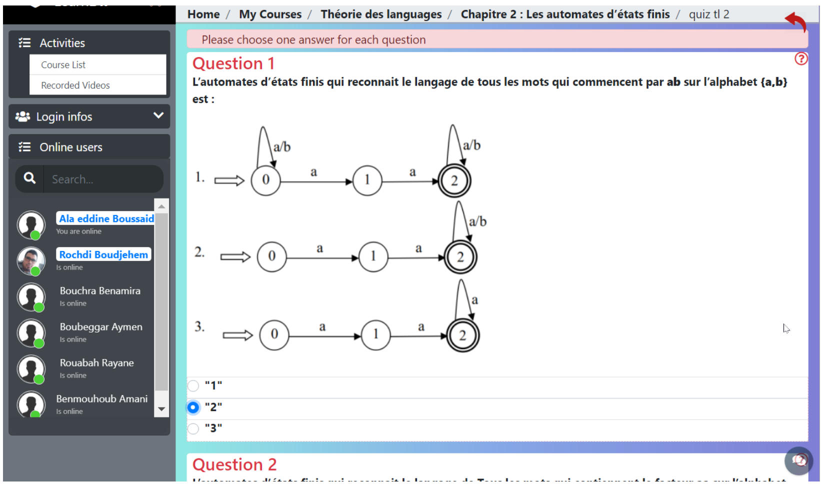


Figure 5.30 – *Evaluation Attempt*

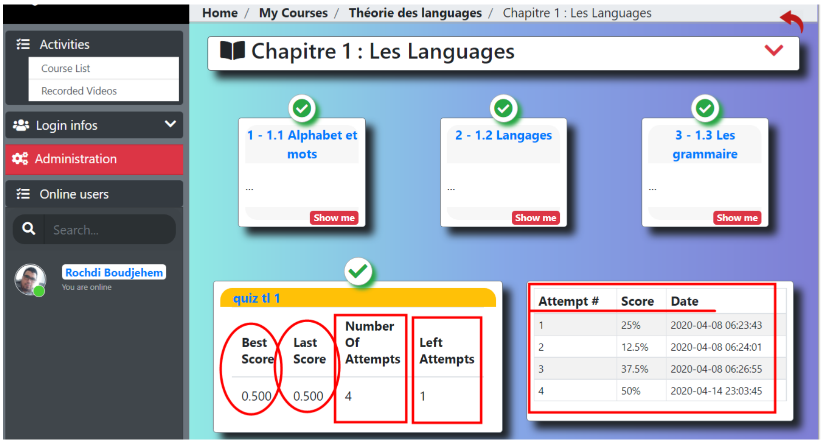


Figure 5.31 – *Evaluation Scores and Attempt Number Check*

Figure 5.32 – *Evaluation Access through the Lesson*

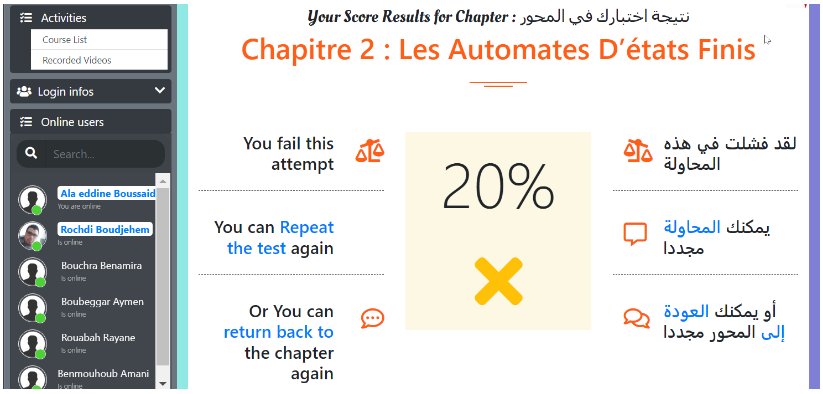


Figure 5.33 – *Evaluation Access through the Chapter*

5.4.3.4 Self Aware Condition Check

The Student can check his Courses' progress and difficulties (Figure.5.34) or choose a specific chapter to see his progress, difficulty level, and evaluation attempts and scores (Figure.5.35)

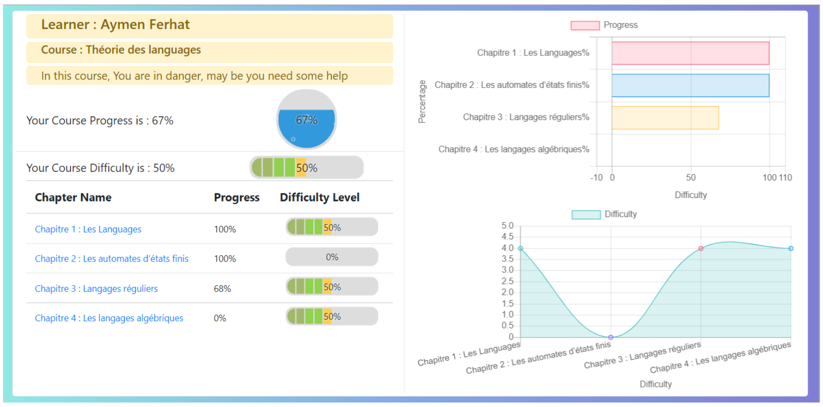


Figure 5.34 – *Course Progress and Difficulties*

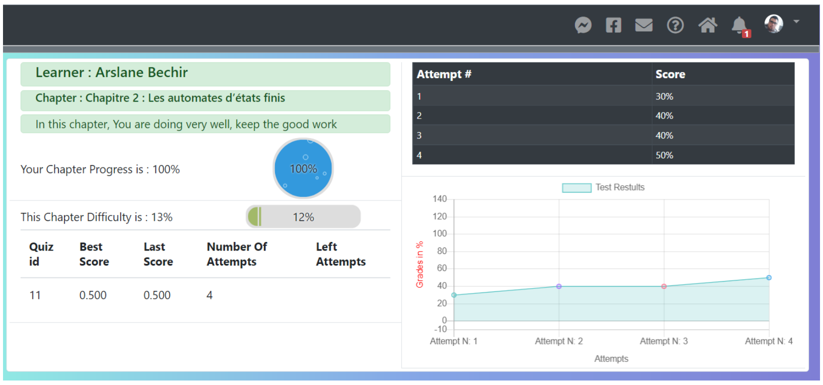


Figure 5.35 – *Chapter Progress, Difficulty Level and Scores*

5.5 Experimentations, Results and Discussion

To validate the proposed approach, we conducted experiments on each project contribution. The first Experminent regarding the first contribution prototype, "LearnDiP," was conducted on second-year computer science students from the University of Guelma in April 2020. The second experiment regarding the second contribution prototype, "LearnDiP+," was conducted on second-year computer science students in May 2021. This section presents these two experiments. Found results are presented in Table. 5.3.

Expt.	Objectives	Total/kept participants Nb.	Used Prototype	Used Tests
1	Testing the effectiveness of the detection and intervention process	52 / 40	LearnDiP	- Kolmogorov-Smirnov - Shapiro-Wilk - Mann-Witney
2	Testing the effectiveness of the prediction and early intervention process	65 / 54	LearnDiP+	- Confusion Matrix

Table 5.3 – *Conducted experimentations.*

5.5.1 Experimentation I: Testing the effectiveness of the prototype to detect at-risk learners and intervene to enhance their cognitive level and diminish dropouts number

To validate the first contribution approach, we experimented with the first contribution prototype, "LearnDiP, on second-year computer science students from the University of Guelma, in April 2020 and during the Coronavirus mandatory home confinement. In this test, we aim to measure the effectiveness of the proposed technique in identifying and assisting at-risk students, thus (1) Reducing the number of dropouts and (2) Improving the cognitive level of students. The experiment was conducted using the first contribution prototype "LearnDiP".

5.5.1.1 Participants

A total of 40 students participated in the experiment. They were randomly divided into two groups of 20 students each. Test Group (TG) consisted of 20 students who got support during the experiment and had access to an informative dashboard to keep them updated on their progress. The second group, known as the Control Group (CG), consisted of 20 students who received no support during the experiment and were entirely ignorant of their condition.

5.5.1.2 Methodology

Students were enrolled in the "Theory of Languages" course, composed of four *LOs*. Each *LO's* evaluation activity consists of a set of Multiple Choice Questions (MCQ) that must be answered correctly before moving on to the next LO when this latter starts. As presented in Formula 5.1, to calculate the Student L_i score in the LO_k evaluation test, we divide the number of correct answers by the total number of questions:

$$Score\,(L_i)_j = \frac{Number\ of\ correct\ answers_j}{Total\ Number\ of\ questions_J} \tag{5.1}$$

to ensure equitable conditions regarding the course quality, the same teacher instructed the course for both groups. Furthermore, neither group has any prior understanding of the course, has never used a distance-learning system, and is utilizing the System for the first time. In addition to calculating the student "Difficulty Levels" for the sake of this experiment, we are also interested in counting the number of dropouts. As a result, this experiment considers a student a "dropout" if he failed to access the last LO, in our case, LO_4. We conducted a "Before/After status" experiment with two randomly assigned independent groups to see the proposed approach's effectiveness. In the first stage, we compare the number of dropouts inside these two groups to check if applying the proposed approach has reduced the number of dropouts. In the next stage, we calculate and compare the two groups' cognitive levels to see any improvement in the TG compared to the CG. The cognitive level in the PreTest and Posttest are given in formulas (5.2) and (5.3) respectively.

$$PreTestCognitiveLevel(L_i) = \frac{Score(L_i)_1 + Score(L_i)_2}{2} \tag{5.2}$$

$$post\text{-}testCognitiveLevel(L_i) = Score(L_i)_3 \tag{5.3}$$

5.5.1.3 Results

Among the enrolled students (52 students), 77% of them (40 students) have seriously attended the course. These maintained students are considered the experiment sample students, while the others are discarded. The maintained sample students were randomly divided into a Test Group (TG) and a Control Group (CG); each group consisted of 20 students. As presented in Table. 5.4, 55% of participant students have dropped out while 45% succeeded. The majority of droppers are from the CG.

	Passed Students		**Dropped Students**	
	Number	**Percentage**	**Number**	**Percentage**
Test Group (TG)	14	78%	6	27%
Control Group (CG)	4	22%	16	73%
Total	18	100%	22	100%

Table 5.4 – *Dropout statistics among TG and CG*

As a result, the number of droppers in the TG is low, indicating that our system LearnDiP's intelligent detection and intervention is effective. However, we must use statistical tests to determine whether or not receiving assistance from LearnDiP is effective and, if so, how significant this improvement is.

A. The Normality Test (Kolmogorov-Smirnov and Shapiro-Wilk): First, we performed normality tests using the following hypotheses to help us decide on the best test to use:

- H_0: The sample data are not significantly different from a normal population.
- H_1: The sample data significantly differ from a normal population.

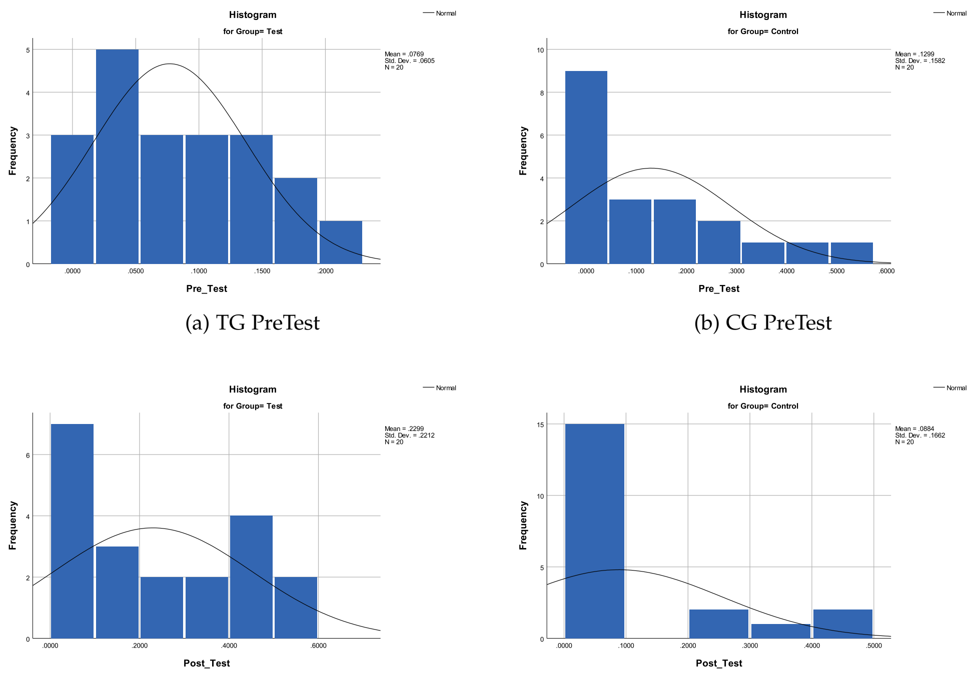
Table. (5.5) displays the results of the normality tests, Kolmogorov-Smirnov and Shapiro-Wilk. Both tests suggest rejecting H_0 just for TG and only during the PreTest. Tests also suggest accepting H_0 for all the other three cases, indicating that majority of our dataset is not normally distributed. (Figure. 5.36).

	Group	**Kolmogorov-Smirnov**			**Shapiro-Wilk**		
		Statistic	**df**	**Sig.**	**Statistic**	**df**	**Sig.**
PreTest	Test	.154	20	**.200**	.928	20	**.139**
	Control	.211	20	.020	.819	20	.002
PostTest	Test	.201	20	.034	.850	20	.005
	Control	.453	20	.000	.585	20	.000

Table 5.5 – *Tests of Normality*

B. The Mann-Whitney U Test (Wilcoxon test for independent samples): As we already mention, most of our samples are not normally distributed. As a result, we chose to use the Mann-Whitney nonparametric U test (Wilcoxon test for independent samples) with the following hypotheses:

- H_0^1: The PreTest scores' distribution is the same across categories of Group (test, control).
- H_1^1: The PreTest scores' distribution is not the same across categories of the Group.



(a) TG PreTest

(b) CG PreTest

(c) TG PostTest

(d) CG PostTest

Figure 5.36 – *Distribution of PreTest and PostTest values for TG and CG*

– H_0^2: The Posttest scores' distribution is the same across categories of the Group.

– H_1^2: The Posttest scores' distribution is not the same across categories of the Group.

The P_{value} of H_0^1 is $P > 0.05$, so we accept H_0^1, stating that there was no significant difference between the cognitive levels of the two groups during the "PreTest" phase (before any intervention of LearnDiP on the TG). On the other hand, the P_{value} of H_0^2 is $P < 0.05$. So, we reject H_0^2 in favor of H_1^2. Therefore, we accept a significant difference between the TG and CG cognitive levels after the intervention of LearnDiP on the TG. All the details of the conducted "U Test" are presented in Table. 5.6.

Total N	40
Mann-Whitney U	192.500
Wilcoxon W	402.500
Test Statistic	192.500
Standard Error	36.608
Standardized Test Statistic	-.205
Asymptotic Sig.(2-sided test)	**.838**
Exact Sig.(2-sided test)	.841

(a) PreTest across Group

Total N	40
Mann-Whitney U	281.500
Wilcoxon W	491.500
Test Statistic	281.500
Standard Error	33.667
Standardized Test Statistic	2.421
Asymptotic Sig.(2-sided test)	**.015**
Exact Sig.(2-sided test)	.026

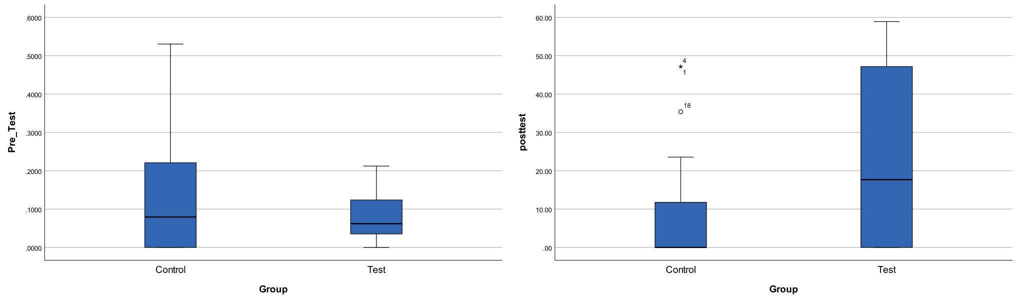
(b) PostTest across Group

Table 5.6 – *Independent-Samples Mann-Whitney U Test Summary PreTest and PostTest across Group*

5.5.1.4 Discussion:

At this point, we could raise some points:

- We believe that the non-normality distribution is a consequence of the enormous

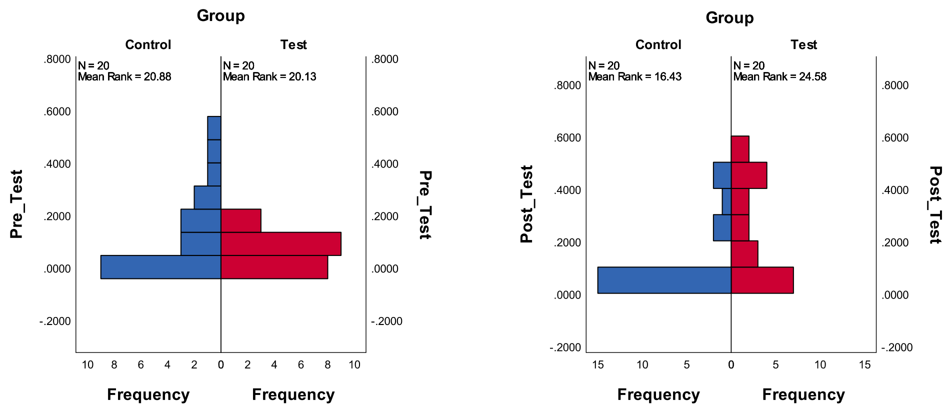


(a) PreTest (b) PostTest

Figure 5.37 – *Differences in means of PreTest and PostTest values for TG and CG*

amount of null scores generated due to the dropouts, making the distribution very skewed and very different from Normal.

- As shown in Figure. 5.37, the posttest TG distribution mean is significantly greater than the CG mean. As a result, the test results show a clear difference in favor of the TG. In addition, Figure. (5.38a) demonstrates that the distribution of CG outcomes was marginally better than TG before the intervention of our approach. . However, the distribution of the TG outcomes became considerably better than that of CG after the intervention of the proposed system, as shown in Figure. (5.38b).



(a) PreTest (b) PostTest

Figure 5.38 – *Independent Samples of the PreTest and PostTest values for TG and CG*

As explained, all test results demonstrate the effectiveness of the proposed method in reducing the number of dropouts and increasing students' cognitive levels through identification and assistance.

5.5.2 Experimentation II: Testing the effectiveness of the prototype to predict struggling learner dropouts

To validate the second contribution's approach, we experimented using the second contribution prototype "LearnDiP+', on Computer Science students from the

University of Guelma in May 2021. In this experiment, we aimed to measure the effectiveness of the proposed technique in (1) Predicting learning difficulties and dropout students and thus (2)Reducing the number of dropout students. The experiment was conducted using the second contribution prototype, "LearnDiP+".

5.5.2.1 Participants

This experiment included 65 second-year computer-science students. Participant students have no prior knowledge of the course and have never used a distance learning system.

The "Theory of Languages" course is available to all enrolled students. It has continuous evaluations distributed weekly throughout the semester. Each evaluation activity consists of a set of Multiple Choice Questions (MCQs). Before moving on to the next $LO_i + 1$, the student must pass the current LO_i evaluation activity.

5.5.2.2 Methodology:

During the initial data analysis, we observed the student's interactions with the proposed system "LearnDiP+" throughout the course period. We discarded data Before analyzing the results, All students who enrolled in the course but did not use any materials were discarded. As a result, we only kept and analyzed the data of 54 students out of a total of 66. The Confusion Matrix technique was used to compare predicted and actual values for each student to determine the quality of the prediction algorithms.

5.5.2.3 Results:

As presented in Table. 5.7, the used matrix provides information about actual and predicted classifications. A positive value means that the student dropped out, while a negative value means that the student graduated. We used four measures for our matrix:

- **True-positive (TP):**
  – The predicted and actual values matched,
  – The actual value is positive (dropout), and the model predicted it as positive.
- **True-negative (TN):**
  – The predicted and actual values matched,
  – The actual value is negative (success), and the model predicted it as negative.
- **False-positive (FP) - Type 1 error:**
  – The predicted and actual values did not match.
  – The actual value is negative, but the model falsely predicted it as positive.

- **False-negative (FN) - Type 2 error:**
  - The predicted and actual values did not match.
  - The actual value is positive, but the model falsely predicted it as negative.

		Actual		
		Positive	**Negative**	**Total**
Predicted	**Positive**	11	12	23
	Negative	1	30	31
	Total	12	42	**54**

Table 5.7 – *Confusion Matrix*

We calculate the "Accuracy" (Formula 5.4), "Recall" (Formula 5.5), "Precision" (Formula 5.6), F1-Score (Formula 5.7) of the Confusion Matrix and presented them in Table. 5.7.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{5.4}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{5.5}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{5.6}$$

$$F1\text{-}Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.7}$$

Total Number	Discarded	Effective	Accuracy	Precision	**Recall**	F1-Score
65	11	54	75.93%	47.8%	**91.7%**	6.29%

Table 5.8 – *Prediction's Confusion matrix*

5.5.2.4 Discussion:

"Precision" is a valuable metric in cases where false positives are more critical than false negatives. Such scenarios can be found in Recommendation Systems or e-Commerce Websites, where a wrong prediction may harm the business or lead to the loss of customers. However, "Recall" is a valuable metric in cases where avoiding False Negatives is more important than avoiding False Positives. In these cases, like the case of our work, raising a false alarm for a graduate student is less harmful than letting an endangered student go undetected, which might drive him to drop out.

In this work, we are more invested in avoiding false negatives, which are cases where the proposed system falsely predicts a dropout student as a graduate; that is why we focus on the "Recall" value to evaluate the proposed system's efficiency. As shown in Table. 5.8, the calculated "Recall" value for the predictive model is 0.917, which is a perfect result, meaning that the model has failed to predict dropout cases only 8% of the time. At the same time, the calculated accuracy of the predictive model is 75.93, which is very acceptable.

5.6 Conclusion

This chapter introduced the two developed prototypes, "LearnDiP" and "LearnDiP+", used to test and validate theoretical approaches presented in the previous chapter. It introduces the different technologies used to develop these prototypes. Moreover, it presents the main functionalities supplied by them to all the actors present in the learning process, learners, teachers, and administrators. It presents each of these actors' spaces and exposes their available functions, and illustrates with some screenshots.

This chapter also presented the conducted experiments using those two prototypes. It highlighted the obtained results through experiments to validate the proposed approaches for detecting, predicting, and preventing learning difficulties in human learning environments.

In this second part, we outlined the choices, concepts, and notions that we addressed in the first place. We tried to combine the psychological and computer science approaches, which is not easy, given that the objectives and methods specific to these fields are relatively distant. However, various proposals have emerged from this work. A new model has been presented and updated to mirror the learner's behavior. Many Indicators have been used to measure the amount of difficulty each learner is facing. That same difficulty level is used to detect and predict the learners who are struggling and are in difficulty.

At the implementation level, in the first contribution, we propose an innovative software architecture, LearnDiP, allowing the modeling and monitoring of learners to understand their behaviors and measure the amount of difficulty they are encountering. In the second contribution, this system receives a significant upgrade and becomes LearnDiP+ capable of making future predictions of struggling learners, providing more time to launch early, targeted and autonomous interventions.

As for the experiments, they support the hypotheses claiming the proposed systems' efficacity and efficiency.

Conclusion and Future Work

When considering the nature of learning, the concept that we acquire knowledge by interacting with our surroundings comes to mind first. When a child plays, waves his arms, or looks around, he has no explicit instructor but a direct sensory link with his surroundings. This relationship reveals a wealth of information regarding cause and effect, the consequences of actions, and how to attain one's objectives. Such interactions are unquestionably an essential source of information about our surroundings and ourselves throughout our lives. Whether learning to drive a car or hold a conversation, we are highly aware of how our environment reacts to our actions and attempt to control what transpires through our behavior. Interactional learning is a fundamental concept underlying all learning and intelligence theories (Sutton and Barto, 2012)

Understanding the learning mechanism is crucial for teachers if they want to succeed in their mission. Teaching learners is a demanding responsibility because of the nature and number of existing barriers and difficulties that could impede the learning operation. These barriers are incarnated in the form of numerous daily difficulties and challenges that learners face. These difficulties affect the learners at several psychological and cognitive levels and can hinder their learning quality.

These difficulties could be provoked or escalated by factors such as inadequate or limited teaching or by inappropriate curriculum content (Kershner, 2000; Robertson et al., 1994);

That is not to say that certain qualities of learners do not predispose them to learning difficulties. Features such as sensory disabilities, limited intelligence, developmental problems, a lack of assistance at home, frequent absences, and other characteristics can all lead to difficulties. Although many learners' features, such as home background, poverty, health, handicap, and intellect, are challenging to change. Still, teaching quality may be enhanced, which may benefit all learners. Rather than looking for deficiencies within a learner, it is usually far more efficient to explore extrinsic factors like the form and relevance of the curriculum and the quality of education the student receives. These variables are more adjustable and flexible to change and improvement than student-specific factors (Westwood, 2006).

Learners with difficulties necessitate different support forms to learn more efficiently and, if possible, catch up with their peers in terms of academic achievement alongside social development. However, it is vital to identify learners having difficulties in the first place before so they can be targeted with the appropriate form of support. Nevertheless, before that, it is imperative to determine the signs and indicators pointing the finger at these learners, especially in e-learning and distance learning environments where there is less or no physical contact between the learners and their instructors. The next step is to identify how best to overcome those difficulties, which involves finding the most significant factors that can be manipulated within the learning environment.

Even though Technology is now more involved in education at all levels, it does not necessarily improve education. Sometimes, it adds more challenges and difficulties for learners and teachers. Learners benefit significantly from technology since it allows them to effortlessly access knowledge from all around the world at any time. As learners become more engaged with new developments, the benefits and drawbacks of increased-technology usage have become apparent.

In that direction, our primary focus in this study is to develop an Early Warning System capable of identifying, predicting, and assisting learners with difficulties inside e-Learning systems. Therefore, it is primordial to understand the nature and the causes of learning difficulties faced by learners overall and specifically in e-learning environments. It is also crucial to investigate the consequences of such difficulties on these learners on several levels if we want to relieve some of these difficulties from their shoulders and thus help these learners carry out their education in better conditions to let them learn better.

By exposing the most significant signs and indicators, we could identify present learning difficulties and those learners facing them. Our research axis is based on constructing a model for each learner, gathering all his interactions with the learning environment that could be used to profile him and be used as a virtual substitute for that physical learner. This model can be seen as the virtual clone of that learner, so when the real learner exhibits any kind of struggle while learning, it may reflect his current and even future state.

We have proposed a new structure for the "Learner Model" by extracting and analyzing his left traces while interacting with the e-learning environment. The proposed model comprises two parts, Static and Dynamic. While the former constitutes basic information about the identity of that learner, the latter constitutes the heart of our contributions. The dynamic part of the "Learner Model" continues to evolve as the learner progresses in his learning and evaluation activities for all his enrolled courses. It is based on a set of "Primary" and "Secondary" calculable indicators tracking his behavioral and cognitive changes and fluctuations during his active presence in the e-learning environment. These indicators are used to calculate a single human-understandable value reflecting his level of difficulty on each single Learning Object and a global "Difficulty Level" for each course that could be used by the system to autonomously intervene to alert the learner and his teachers about potential learning difficulties.

Moreover, the set of Learning Objects' Difficulty Levels for each course represents a linear vector that we have called the "Course Difficulty Vector" (CDV), representing the learners' ups and downs during that course. This vector could be seen as the learner's digital fingerprint of his faced difficulties during that course. By comparing the current learner's CDV with previous learners' ones, we could find similar patterns from the past. Using these similar patterns, we could predict the future learning difficulties state of that learner.

The detection process starts by building a model for each learner based on his gathered traces. These models are used afterward to establish a difficulty level for each enrolled course by this learner. These models also store the learner's difficulty history to be used in the prediction process alongside his peer's models. Detected or predicted struggling learners are taken care of by the autonomous intervention process.

In the proposed approach, four abstract subsystems in constant collaboration identify, predict and rescue endangered learners. The "Learner Model Update System" builds and updates the learners' models. While the "Difficulty Detection Subsystem" tracks and monitors the learners and keeps updating their primary and secondary indicators and difficulty levels to assess their difficulty status for each course. On the other hand, the "Difficulty Prediction Subsystem" compares the current learner's CDV with his previous peers to find similar patterns and predict the learners' difficulties status. The "Intervention Subsystem" stays in standby mode, waiting for any detected or predicted difficulties to take the necessary measures.

As we explained, the theoretical proposed approach may seem simple. However, a realistic e-learning system has hundreds to thousands of learners and dozens to hundreds of courses and teachers. How can any of the proposed subsystems keep track of all the learners and courses simultaneously? One solution imposes itself, which is using Distributed Artificial Intelligence (DAI) and, precisely, its most known form, Multi-Agent Systems (MAS). In that direction, each of these proposed subsystems delegates a set of cognitive agents to carry out its designed tasks. Some of these agents are assigned per learner, like the "Personal Agent", which needs to monitor each learner individually, while others are assigned per course.

In the context of this research and to validate the theoretical proposed approaches, we build two successive complementary prototypes called "LearnDiP" and "LearnDiP+." The two prototypes were built as PHP-scripting-language web-based applications that were deployed on the Internet to be accessible to students. We performed two experiments with higher education students from the University of Guelma. The first experiment was conducted with second-year computer-science students in April 2020, and the second was conducted with second-year computer-science students in May 2021.

During these experiments, we faced many challenges, mainly because of the quality of the Internet and "Information and Communication Technologies" (ICTs). Problems like the ICT hardware availability, quality, and price make procuring decent and reliable hardware very hard for students and teachers. Furthermore, with low Internet flow and continuous cut-offs in landline and mobile Internet connections, reliable Internet is still a luxury in the country where the experiments were conducted.

The experiment results supported the efficiency of the proposed EWS. They demonstrated the importance of using our primary and secondary indicators to detect, predict, and rescue learners at early stages, thus reducing the number of dropout learners. Unfortunately, we could not compare our prediction results with any reviewed works for similar reasons because we did not experiment on the same groups or in the same conditions.

In future work, we intend to enhance the "Learner Model" by refining chosen indicators by finding new indicators and eliminating less influential ones based on conducted experiments' results.

We also intend to enrich the prediction process with other techniques to have multiple predictions and ensure that no learner with difficulties is left alone.

One of the limitations of our proposed approach is the intervention subsystem which did not have enough share of thinking. We intend to transform it into a more robust "recommender system" and empower it with more tools to widen its reach range and ensure more adapted content for the learners.

Author's Publications

Several parts of this research have been submitted and mostly published in international journals and conferences:

1. Boudjehem, R., Lafifi, Y. (2021). A new approach to identify dropout learners based on their performance-based behavior. *JUCS - Journal of Universal Computer Science*, 27(10), 1001-1025, https://doi.org/10.3897/jucs.74280
2. Benyounes, A., Boudjehem, R., Lafifi, Y. (2020). Study of the impact of collaboration among learners during the learning of "Object-Oriented Programming." In S. Gülseçen, M. Gezer, F. Bayraktutan (Eds.), *FL2020, 8th International Conference on Future Learning and Informatics: "Data Revolution"* (pp. 55-56). Istanbul University Press Istanbul, Turkey.
3. Yacine, L., Boudjehem, R., Benoughiden, R., Mehnaoui, Z. (2019). LISP(P): A new pedagogical approach for learning mathematics in Colleges. *CITCS'2019, 1st International Conference on Innovative Trends in Computer Science*, 25.
4. Boudjehem, R., Lafifi, Y. (2019). A decision support system to assist course design teachers. *IAM'19, 2nd Conference On Informatics And Applied Mathematics*.
5. Boudjehem, R., Lafifi, Y. (2018). Agent-based solutions in distance education. *IAM'18 Doctoral Days on Informatics and Applied Mathematics*, 33-39.

Bibliography

Abe, J. A. A. (2020). Big five, linguistic styles, and successful online learning. *The Internet and Higher Education*, *45*, 100724, https://doi.org/10.1016/j.iheduc.2019.100724.

Abioye, S. O., et al. (2021). Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges. *Journal of Building Engineering*, *44*(2021), 103299, https://doi.org/10.1016/j.jobe.2021.103299.

Abood, M. H., Alharbi, B. H., Mhaidat, F., & Gazo, A. M. (2020). The Relationship between Personality Traits, Academic Self-Efficacy and Academic Adaptation among University Students in Jordan. *International Journal of Higher Education*, *9*(3), 120, https://doi.org/10.5430/ijhe.v9n3p120.

Abosi, O. (2007). Educating Children with Learning Disabilities in Africa. *Learning Disabilities Research & Practice*, *22*(3), 196–201, https://doi.org/10.1111/j.1540-5826.2007.00242.x.

Abu Naser, S. S. (2016). ITSB: An Intelligent Tutoring System Authoring Tool. *Available online www.jsaer.com Journal of Scientific and Engineering Research*, *63*(5), 63–71.

Aggarwal, D., Mittal, S., & Bali, V. (2021). Significance of Non-Academic Parameters for Predicting Student Performance Using Ensemble Learning Techniques. *International Journal of System Dynamics Applications*, *10*(3), 38–49, https://doi.org/10.4018/ijsda.2021070103.

Agostinelli, M. D. J. (2019). *From Distance Education To Online Education: A Review of the Literature*. Technical report, University of Montana, Missoula, MT.

AL-Malaise, A., Malibari, A., & Alkhozae, M. (2014). Students Performance Prediction System Using Multi Agent Data Mining Technique. *International Journal of Data Mining & Knowledge Management Process*, *4*(5), 01–20, https://doi.org/10.5121/ijdkp.2014.4501.

Alban-Metcalfe, J. & Alban-Metcalfe, J. (2001). *Managing Attention Deficit/Hyperactivity Disorder in the Inclusive Classroom*. David Fulton Publishers, 1st edition.

Albatish, I., Mosa, M. J., & Abu-Naser, S. S. (2018). ARDUINO Tutor: An Intelligent Tutoring System for Training on ARDUINO. *International Journal of Engineering and Information Systems (IJEAIS)*, *2*(1), 236–245.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences*, *4*(2), 167–207, https://doi.org/10.1207/s15327809jls0402_2.

Atkinson, R. K. (2002). Optimizing learning from examples using animated pedagogical agents. *Journal of Educational Psychology*, *94*(2), 416–427, https://doi.org/10.1037/0022-0663.94.2.416.

Atkinson, R. K., Mayer, R. E., & Merrill, M. M. (2005). Fostering social agency in multimedia learning: Examining the impact of an animated agent's voice. *Contemporary Educational Psychology*, *30*(1), 117–139, https://doi.org/10.1016/j.cedpsych.2004.07.001.

Azcona, D. & Casey, K. (2015). Micro-analytics for student performance prediction leveraging fine-grained learning analytics to predict performance. *International Journal of Computer Science and Software Engineering*, *4(8)*(8), 218–223.

Azevedo, R. & Hadwin, A. F. (2005). Scaffolding Self-regulated Learning and Metacognition – Implications for the Design of Computer-based Scaffolds. *Instructional Science*, *33*(5-6), 367–379, https://doi.org/10.1007/s11251-005-1272-9.

Badian, N. A. (1996). Dyslexia: A validation of the concept at two age levels. *Journal of Learning Disabilities*, *29*(1), 102–112, https://doi.org/10.1177/002221949602900113.

Bakhta, N. (2014). *Modèle multi-agents pour la conception de systèmes d ' aide à la décision collective*. PhD thesis, University of Oran.

Bandura, A., Freeman, W. H., & Lightsey, R. (1999). Self-Efficacy: The Exercise of Control. *Journal of Cognitive Psychotherapy*, *13*(2), 158–166, https://doi.org/10.1891/0889-8391.13.2.158.

Baneres, D., Rodriguez, M. E., & Serra, M. (2019). An Early Feedback Prediction System for Learners At-Risk Within a First-Year Higher Education Course. *IEEE Transactions on Learning Technologies*, *12*(2), 249–263, https://doi.org/10.1109/TLT.2019.2912167.

Baum, S. D., Barrett, A. M., & Yampolskiy, R. V. (2017). Modeling and interpreting expert disagreement about artificial superintelligence. *Informatica (Slovenia)*, *41*(4), 419–427, https://doi.org/10.31449/inf.v41i4.1812.

Baylor, A. L. (2009). Promoting motivation with virtual agents and avatars: role of visual presence and appearance. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1535), 3559–3565, https://doi.org/10.1098/rstb.2009.0148.

Belkhadir, I., Omar, E. D., & Boumhidi, J. (2019). An intelligent recommender system using social trust path for recommendations in web-based social networks. In *Procedia Computer Science*, volume 148 (pp. 181–190).: Elsevier B.V.

Bender, W. N. (2004). *Learning disabilities: Characteristics, identifi cation, and teaching strategies.* Boston: Allyn & Bacon, 5th edition.

Benyounes, A., Boudjehem, R., & Lafifi, Y. (2020). Study of the impact of collaboration among learners during the learning of "Object-Oriented Programming". In *8th International Conference on Future Learning and Informatics FL2020", Istanbul University, , Turkey.* (pp. 55–56).

Bernstein, S., Atkinson, A. R., & Martimianakis, M. A. (2013). Diagnosing the learner in difficulty. *Pediatrics*, *132*(2), 210–212, https://doi.org/10.1542/peds.2013-1526.

Bersin, J. (2004). *The Blended Learning Book: Best Practices, Proven Methodologies, And Lessons Learned*. San Francisco, USA: Pfeiffer, 1st edition.

Biggs, J. (1995). Motivating learning. In J. Biggs & D. Watkins (Eds.), *Classroom learning* (pp. 82–102). Singapore.: Prentice Hall.

Blake, R. J. (2009). Brave new digital classroom: technology and foreign language learning. *Choice Reviews Online*, *46*(11), 46–6326–46–6326, https://doi.org/10.5860/CHOICE.46-6326.

Bodily, R. & Verbert, K. (2017). Review of Research on Student-Facing Learning Analytics Dashboards and Educational Recommender Systems. *IEEE Transactions on Learning Technologies*, *10*(4), 405–418, https://doi.org/10.1109/TLT.2017.2740172.

Boudjehem, R. & Lafifi, Y. (2021). A new approach to identify dropout learners based on their performance-based behavior. *JUCS - Journal of Universal Computer Science*, *27*(10), 1001–1025, https://doi.org/10.3897/jucs.74280.

Brahim, B. & Lotfi, A. (2020). A traces based system helping to assess knowledge level in e-learning system. *Journal of King Saud University - Computer and Information Sciences*, *32*(8), 977–986, https://doi.org/10.1016/j.jksuci.2018.10.008.

Brandon, W. R. (2008). *Best of the eLearning Guild's learning solutions: top articles from the eMagazine's first five years*. San Francisco, California: Pfeiffer.

Bratko, D., Chamorro-Premuzic, T., & Saks, Z. (2006). Personality and school performance: Incremental validity of self- and peer-ratings over intelligence. *Personality and Individual Differences*, *41*(1), 131–142, https://doi.org/10.1016/j.paid.2005.12.015.

Brennan, W. K. (1985). *Curriculum for special needs*. Open University Press.

Brooks, C., Greer, J., Melis, E., & Ullrich, C. (2006). Combining ITS and eLearning Technologies: Opportunities and Challenges. In M. Ikeda, K. D. Ashley, & T.-W. Chan (Eds.), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Lecture Notes in Computer Science (pp. 278–287). Berlin, Heidelberg: Springer.

Buche, C. (2005). *Un système tutoriel intelligent et adaptatif pour l'apprentissage de compétences en environnement virtuel de formation*. PhD thesis, Université de Bretagne occidentale - Brest, 2005.

Burns, H. L. & Capps, C. G. (1988). Foundations of Intelligent Tutoring Systems: An Introduction. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 1–20). Hillsdale, New Jersey: Lawrence Erlbaum Associates Publishers.

Butz, C. J., Hua, S., & Maguire, R. B. (2006). A web-based bayesian intelligent tutoring system for computer programming. *Web Intelligence and Agent Systems*, *4*(1), 61–81.

Campbell, F. A. & Ramey, C. T. (1994). Effects of Early Intervention on Intellectual and Academic Achievement: A Follow-up Study of Children from Low-Income Families. *Child Development*, *65*(2), 684–698, https://doi.org/10.1111/j.1467-8624.1994.tb00777.x.

Çano, E. & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, *21*(6), 1487–1524, https://doi.org/10.3233/IDA-163209.

Carlyon, T. & Opperman, A. (2020). Embedding Soft Skills Within Learner-Centred Environments For Vocation Education Graduates. *Scope: Contemporary Research Topics (Learning & Teaching)*, *2020*(9), 16–23, https://doi.org/10.34074/scop.4009001.

Casamayor, A., Amandi, A., & Campo, M. (2009). Intelligent assistance for teachers in collaborative e-learning environments. *Computers & Education*, *53*(4), 1147–1154, https://doi.org/10.1016/j.compedu.2009.05.025.

Casey, K. & Azcona, D. (2017). Utilizing student activity patterns to predict performance. *International Journal of Educational Technology in Higher Education*, *14*, 4, https://doi.org/10.1186/s41239-017-0044-3.

Caspi, A., Chajut, E., Saporta, K., & Beyth-Marom, R. (2006). The influence of personality on social participation in learning environments. *Learning and Individual Differences*, *16*(2), 129–144, https://doi.org/10.1016/j.lindif.2005.07.003.

Chambers, J. A. & Sprecher, J. W. (1983). *Computer-assisted instruction: Its use in the classroom*. Upper Saddle River, NJ: Prentice Hall, Upper Saddle River.

Chan, L. K. & Van Kraayenoord, C. E. (1998). Learning through dialogues for students with learning difficulties. *Australian Journal of Learning Disabilities*, *3*(1), 21–26, https://doi.org/10.1080/19404159809546552.

Cheng, P. (1998). Primary teachers' perceptions and understanding of learning difficulties. In D. Chan (Ed.), *Helping students with learning difficulties* (pp. 121–131). Hong Kong: Chinese University Press.

Chung, C. & Ackerman, D. (2015). Student Reactions to Classroom Management Technology: Learning Styles and Attitudes Toward Moodle. *Journal of Education for Business*, *90*(4), 217–223, https://doi.org/10.1080/08832323.2015.1019818.

Clancey, W. J., Bennett, J. S., & Cohen, P. R. (1979a). *Applications-Oriented AI Research: Education*. Technical report, Stanford University.

Clancey, W. J., Shortliffe, E. H., & Buchanan, B. G. (1979b). Intelligent Computer-Aided Instruction For Medical Diagnosis. In *Institution of Mechanical Engineers, Conference Publications* (pp. 175–183).

Clarebout, G., Elen, J., Johnson, W. L., & Shaw, E. (2002). Animated pedagogical agents: An opportunity to be grasped? *Journal of Educational Multimedia and Hypermedia*, *11*(3), 267–286.

Clarebout, G. & Heidig, S. (2012). Pedagogical Agents. In *Encyclopedia of the Sciences of Learning* (pp. 2567–2571). Boston, MA: Springer US.

Clemens, C., Kumar, V., Boulanger, D., Seanosky, J., & Kinshuk (2018). Learning Traces, Competence Assessment, and Causal Inference for English Composition. In J. M. Spector, V. Kumar, A. Essa, Y.-M. Huang, R. Koper, R. A. W. Tortorella, T.-W. Chang, Y. Li, & Z. Zhang (Eds.), *Frontiers of Cyberlearning*, Lecture Notes in Educational Technology (pp. 49–67). Singapore: Springer Singapore.

Cohen, P. R. & Levesque, H. J. (1988). *Rational Interaction as the Basis for Communication*. Technical report, Stanford University, University of Toronto.

Colace, F., Santo, M. D., Lombardi, M., Pascale, F., Pietrosanto, A., & Lemma, S. (2018). Chatbot for E-Learning: A Case of Study. *International Journal of Mechanical Engineering and Robotics Research*, *7*(5), 528–533, https://doi.org/10.18178/ijmerr.7.5.528-533.

Cole, D. (2020). The Chinese Room Argument (Stanford Encyclopedia of Philosophy). In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy (Winter 2020 Edition)*. Retrieved from https://plato.stanford.edu/archives/spr2020/entries/chinese-room/.

Dale, V. H. M., McEwan, M., & Bohan, J. (2021). Early adopters versus the majority: Characteristics and implications for academic development and institutional change. *Journal of Perspectives in Applied Academic Practice*, *9*(2), 54–67, https://doi.org/10.14297/jpaap.v9i2.483.

Denden, M., Tlili, A., Essalmi, F., Jemni, M., Chang, M., Kinshuk, & Huang, R. (2019). *iMoodle: An Intelligent Gamified Moodle to Predict "at-risk" Students Using Learning Analytics Approaches*, (pp. 113–126). Springer: Singapore.

Deng, R., Benckendorff, P., & Gannaway, D. (2020). Learner engagement in MOOCs: Scale development and validation. *British Journal of Educational Technology*, *51*(1), 245–262, https://doi.org/10.1111/bjet.12810.

Dettori, G. & Ott, M. (2006). Looking beyond the performance of grave underachievers in mathematics. *Intervention in School and Clinic*, *41*(4), 201–208, https://doi.org/10.1177/10534512060410040201.

Dolenc, K. & Aberšek, B. (2015). TECH8 intelligent and adaptive e-learning system: Integration into Technology and Science classrooms in lower secondary schools. *Computers & Education*, *82*, 354–365, https://doi.org/10.1016/J.COMPEDU.2014.12.010.

Domagk, S. (2010). Do Pedagogical Agents Facilitate Learner Motivation and Learning Outcomes? *Journal of Media Psychology*, *22*(2), 84–97, https://doi.org/10.1027/1864-1105/a000011.

Dowker, A. (2004). *What works for children with mathematical difficulties?* Technical report, University of Oxford.

Drogoul, A. (1993). *De La Simulation Multi-Agent A La Résolution Collective de Problèmes: Une Étude De l'Émergence De Structures D'Organisation Dans Les Systèmes Multi-Agents De la Simulation Multi-Agents à la Résolution Collective de Problèmes: une Etude de l'Emergence de S*. PhD thesis, University of Paris VI.

DuPaul, G.J. & Stoner, G. (2003). *ADHD in the schools: Assessment and intervention strategies*. New York: Guilford Press.

Durfee, E. H., Lesser, V. R., & Corkill, D. D. (1987). Coherent Cooperation Among Communicating Problem Solvers. *IEEE Transactions on Computers*, *C-36*(11), 1275–1291, https://doi.org/10.1109/TC.1987.5009468.

Elaachak, L., Belahbibe, A., & Bouhorma, M. (2015). Towards a System of Guidance, Assistance and Learning Analytics Based on Multi Agent System Applied on Serious Games. *International Journal of Electrical and Computer Engineering (IJECE)*, *5*(2), 344, https://doi.org/10.11591/ijece.v5i2.pp344-354.

Elkins, J. (2007). Learning disabilities: Bringing fields and nations together. *Journal of Learning Disabilities*, *40*(5), 392–399, https://doi.org/10.1177/00222194070400050201.

Elliott, P. & Garnett, C. (1994). Mathematics power for all. In C. A. Thornton & N. S. Bley (Eds.), *Windows of Opportunity: Mathematics for Students with Special Needs.* Victoria, Australia: National Council of Teachers of Mathematics Reston Inc.,.

Erickson, T. (1997). Designing agents as if people mattered. *Software agents*, (pp. 79–96).

Faibisoff, S. G. & Willis, D. J. (1987). Distance Education: Definition and Overview. *Journal of Education for Library and Information Science*, 27(4), 223, https://doi.org/10.2307/40323650.

Farkota, R. M. (2005). Basic math problems: the brutal reality! *Learning Difficulties Australia Bulletin*, 37(3), 10–11.

Fellmann, M., Lambusch, F., & Dehne, M. (2020). Towards intelligent personal task and time management: Requirements and opportunities for advanced to-do lists. In *CEUR Workshop Proceedings*, volume 2628 (pp. 25–31).

Fenza, G. & Orciuoli, F. (2016). Building Pedagogical Models by Formal Concept Analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 144–153). Springer Verlag.

Ferber, J. (1997). *Les systèmes multi-agents: vers une intelligence collective.* InterEditions.

Ferber, J. & Weiss, G. (1999). *Multi-Agent System: An Introduction to Distributed Artificial Intelligence*. MIT Press.

Finn, J. D. & Wong, K. K. (2002). *Taking small classes one step further*. IAP.

Fischetti, E. & Gisolfi, A. (1990). From computer-aided instruction to intelligent tutoring systems. *Educational Technology*, *30*(8), 7–17, https://doi.org/10.5555/92704.92705.

Flexible Learning Advisory Group (2004). *2005 Australian Flexible Learning Framework for the national vocational education and training system. 2005 Framework business plan.* Technical report, Flexible Learning Advisory Group, Brisbane, Queensland.

Fox, G. (2003). *A handbook for learning support assistants.* London: Fulton, 2nd edition.

Franklin, S. (2007). A Foundational Architecture for Artificial General Intelligence. In B. Goertzel & P. Wang (Eds.), *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms* (pp. 36–54). Amsterdam: IOS Press.

Freedman, R., Ali, S. S., & McRoy, S. (2000). What Is An Intelligent Tutoring System? *intelligence*, *11*(3), 15–16, https://doi.org/10.1145/350752.350756.

Fry, E. (1960). Teaching Machine Dichotomy: Skinner vs. Pressey. *Psychological Reports*, *6*(1), 11–14, https://doi.org/10.2466/pr0.1960.6.1.11.

Furnham, A., Zhang, J., & Chamorro-Premuzic, T. (2005). The Relationship between Psychometric and Self-Estimated Intelligence, Creativity, Personality and Academic Achievement. *Imagination, Cognition and Personality*, *25*(2), 119–145, https://doi.org/10.2190/530V-3M9U-7UQ8-FMBG.

Gage, N. L. & Berliner, D. C. (1998). *Educational psychology*. Boston: Houghton Mifflin, 6th edition.

Galbraith, A. & Alexander, J. (2005). Literacy, self-esteem and locus of control. *Support for Learning*, *20*(1), 28–34, https://doi.org/10.1111/j.0268-2141.2005.00357.x.

García-Avilés, J. A. (2020). Diffusion of Innovation. In *The International Encyclopedia of Media Psychology* (pp. 1–8). Wiley.

Gasser, L. & Briot, J.-P. (1992). Object-Based Concurrent Programming and Distributed Artificial Intelligence. In N. M. Avouris & L. Gasser (Eds.), *Distributed Artificial Intelligence: Theory and Praxis* (pp. 81–107). Springer Netherlands.

Goldberg, M. (2002). *15 school questions and discussion*. Lanham, MD: Scarecrow Press, Incorporated.

Graesser, A. C. & McNamara, D. (2010). Self-Regulated Learning in Learning Environments With Pedagogical Agents That Interact in Natural Language. *Educational Psychologist*, *45*(4), 234–244, https://doi.org/10.1080/00461520.2010.515933.

Guessoum, Z. (1996). *Un environnement opérationnel de conception et de réalisation de systèmes multi-agents*. PhD thesis, Université Pierre et Marie Curie.

Gulz, A. (2004). Benefits of virtual characters in computer based learning environments: Claims and evidence. *International Journal of Artificial Intelligence in Education*, *14*(3, 4), 313–334.

Hastings, N. & Schwieso, J. (1995). Tasks and tables: the effects of seating arrangements on task engagement in primary classrooms. *Educational Research*, *37*(3), 279–291, https://doi.org/10.1080/0013188950370306.

Heidig, S. & Clarebout, G. (2011). Do pedagogical agents make a difference to student motivation and learning? *Educational Research Review*, *6*(1), 27–54, https://doi.org/10.1016/j.edurev.2010.07.004.

Henderson, R. W. (2002). Queensland Year 2 Diagnostic Net and teachers' explanations of literacy failure. *Australian Journal of Education*, *46*(1), 50–64, https://doi.org/10.1177/000494410204600105.

Hensley, L. C., Wolters, C. A., Won, S., & Brady, A. C. (2018). Academic Probation, Time Management, and Time Use in a College Success Course. *Journal of College Reading and Learning*, *48*(2), 105–123, https://doi.org/10.1080/10790195.2017.1411214.

Holstein, K., Hong, G., Tegene, M., McLaren, B. M., & Aleven, V. (2018). The classroom as a dashboard. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge* (pp. 79–88). New York, NY, USA: ACM.

Holstein, K., McLaren, B. M., & Aleven, V. (2019). Designing for Complementarity: Teacher and Student Needs for Orchestration Support in AI-Enhanced Classrooms. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11625 LNAI (pp. 157–171). Springer Verlag.

Hook, J., El-Sedky, S., De Silva, V., & Kondoz, A. (2021). Learning data-driven decision-making policies in multi-agent environments for autonomous systems. *Cognitive Systems Research*, *65*, 40–49, https://doi.org/10.1016/j.cogsys.2020.09.006.

Howard, E., Meehan, M., & Parnell, A. (2018). Contrasting prediction methods for early warning systems at undergraduate level. *The Internet and Higher Education*, *37*, 66–75, https://doi.org/10.1016/j.iheduc.2018.02.001.

Howe, M. J. A. (1999). *A Teacher ′ s Guide to the Psychology of Learning*. Blackwell: Wiley, 2nd edition.

Huang, S. & Fang, N. (2013). Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models. *Computers & Education*, *61*(1), 133–145, https://doi.org/10.1016/j.compedu.2012.08.015.

Hulme, C. & Mackenzie, S. (2014). *Working Memory and Severe Learning Difficulties (PLE: Memory)*. New York: Psychology Press.

Hussain, M., Zhu, W., Zhang, W., & Abidi, S. M. R. (2018). Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores. *Computational Intelligence and Neuroscience*, *2018*, 1–21, https://doi.org/10.1155/2018/6347186.

Jaggars, S. S. & Xu, D. (2016). How do online course design features influence student performance? *Computers & Education*, *95*, 270–284, https://doi.org/10.1016/j.compedu.2016.01.014.

Jayaprakash, S. M., Moody, E. W., Lauría, E. J., Regan, J. R., & Baron, J. D. (2014). Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative. *Journal of Learning Analytics*, *1*(1), 6–47, https://doi.org/10.18608/jla.2014.11.3.

Jenkins, J. R., Antil, L. R., Wayne, S. K., & Vadasy, P. F. (2003). How cooperative learning works for special education and remedial students. *Exceptional Children*, *69*(3), 279–292, https://doi.org/10.1177/001440290306900302.

Jennings, N. R. & Wooldridge, M. (1998). Applications of Intelligent Agents. In N. R. Jennings & M. J. Wooldridge (Eds.), *Agent Technology* (pp. 3–28). Berlin, Heidelberg: Springer.

Karande, S., Sawant, S., Kulkarni, M., Kanchan, S., & Sholapurwala, R. (2005). Cognition in specific learning disability. *Indian Journal of Pediatrics*, *72*(12), 1029–1033, https://doi.org/10.1007/BF02724405.

Kavale, K. A., Holdnack, J. A., & Mostert, M. P. (2006). Responsiveness to intervention and the identification of specific learning disability: A critique and alternative proposal. *Learning Disability Quarterly*, *29*(2), 113–127, https://doi.org/10.2307/30035539.

Kershner, R. (2000). Teaching children whose progress in learning is causing. In D. Whitebread (Ed.), *The psychology of teaching and learning in the primary school* (pp. 277–299). London, UK: Routledge-Falmer.

Khamparia, A. & Pandey, B. (2020). Association of learning styles with different e-learning problems: a systematic review and classification. *Education and Information Technologies*, *25*(2), 1303–1331, https://doi.org/10.1007/s10639-019-10028-y.

Khin, N. N. & Soe, K. M. (2020). University Chatbot using Artificial Intelligence Markup Language. In *2020 IEEE Conference on Computer Applications(ICCA)* (pp. 1–5).: IEEE.

Klassen, R. M. & Lynch, S. L. (2007). Self-efficacy from the perspective of adolescents with LD and their specialist teachers. *Journal of Learning Disabilities*, *40*(6), 494–507, https://doi.org/10.1177/00222194070400060201.

Kokoç, M. & Altun, A. (2021). Effects of learner interaction with learning dashboards on academic performance in an e-learning environment. *Behaviour and Information Technology*, *40*(2), 161–175, https://doi.org/10.1080/0144929X.2019.1680731.

Koprinska, I., Stretton, J., & Yacef, K. (2015). Students at Risk : Detection and Remediation. In *Proceeding of the 8th International Conference on Educational Data Mining* (pp. 512–515).

Krämer, N. C. & Bente, G. (2010). Personalizing e-Learning. The Social Effects of Pedagogical Agents. *Educational Psychology Review*, *22*(1), 71–87, https://doi.org/10.1007/s10648-010-9123-x.

Kuzilek, J., Hlosta, M., Herrmannova, D., Zdrahal, Z., Wolff, A., & Item, J. (2015). OU analyse : analysing at-risk students at The Open University. *Learning Analytics Review, LAK15 '1*, (pp. 1–16).

Lafifi, Y., Boudjehem, R., Benoughiden, R., & Mehnaoui, Z. (2019). LISP(P) : A new pedagogical approach for learning mathematics in Colleges. In *1st International Conference on Innovative Trends in Computer Science (CITCS'2019)*.

Lakhal, S. & Khechine, H. (2021). Technological factors of students' persistence in online courses in higher education: The moderating role of gender, age and prior online course experience. *Education and Information Technologies*, *26*(3), 3347–3373, https://doi.org/10.1007/s10639-020-10407-w.

Lancaster, J. (2005). *Refereed Papers Is It Really Possible? Can Students With Learning Difficulties Ever Achieve Higher Levels Of Self-Efficacy?* Technical Report 2, School ofTeacher Education, Charles Sturt University.

Lane, H. C., Cahill, C., Foutz, S., Auerbach, D., Noren, D., Lussenhop, C., & Swartout, W. (2013). The Effects of a Pedagogical Agent for Informal Science Education on Learner Behaviors and Self-efficacy. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik

(Eds.), *Artificial Intelligence in Education*, Lecture Notes in Computer Science (pp. 309–318). Berlin, Heidelberg: Springer Berlin Heidelberg.

Larreamendy-Joerns, J. & Leinhardt, G. (2006). Going the Distance With Online Education. *Review of Educational Research*, *76*(4), 567–605, https://doi.org/10.3102/00346543076004567.

Latham, A. M., Crockett, K. A., McLean, D. A., Edmonds, B., & O'Shea, K. (2010). Oscar: An intelligent conversational agent tutor to estimate learning styles. In *International Conference on Fuzzy Systems* (pp. 1–8).: IEEE.

Leiding, D. (2002). *The Won't Learners: An Answer to Their Cry*. Scarecrow Press, Incorporated.

Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., & Bhogal, R. S. (1997). The persona effect: affective impact of animated pedagogical agents. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (pp. 359–366).

Leung, C., Lindsay, G., & Lo, S. K. (2007). Early identification of primary school students with learning difficulties in Hong Kong: The development of a checklist. *European Journal of Special Needs Education*, *22*(3), 327–339, https://doi.org/10.1080/08856250701430919.

Liang, C. C., Liang, W. Y., & Tseng, T. L. (2019). Evaluation of intelligent agents in consumer-to-business e-Commerce. *Computer Standards & Interfaces*, *65*, 122–131, https://doi.org/10.1016/J.CSI.2019.03.002.

Liu, X., Bonk, C. J., Magjuka, R. J., Lee, S.-H., & Su, B. (2005). Exploring Four Dimensions Of Online Instructor Roles: A Program Level Case Study. *Online Learning*, *9*(4), https://doi.org/10.24059/olj.v9i4.1777.

Lu, C. & Wang, S. (2020). The General-Purpose Intelligent Agent.

Lykourentzou, I., Giannoukos, I., Nikolopoulos, V., Mpardis, G., & Loumos, V. (2009). Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers and Education*, *53*(3), 950–965, https://doi.org/10.1016/j.compedu.2009.05.010.

Lyle, S. (1996). An analysis of collaborative group work in the primary school and the factors relevant to its success. *Language and Education*, *10*(1), 13–31, https://doi.org/10.1080/09500789608666698.

Maldonado, S., Miranda, J., Olaya, D., Vásquez, J., & Verbeke, W. (2021). Redefining profit metrics for boosting student retention in higher education. *Decision Support Systems*, *143*(January), https://doi.org/10.1016/j.dss.2021.113493.

Mallinson, D. J. (2021). Policy Innovation Adoption Across the Diffusion Life Course. *Policy Studies Journal*, *49*(2), 335–358, https://doi.org/10.1111/psj.12406.

Maqtary, N., Mohsen, A., & Bechkoum, K. (2019). Group Formation Techniques in Computer-Supported Collaborative Learning: A Systematic Literature Review. *Technology, Knowledge and Learning*, *24*(2), 169–190, https://doi.org/10.1007/s10758-017-9332-1.

Marbouti, F., Diefes-Dux, H. A., & Madhavan, K. (2016). Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education*, *103*, 1–15, https://doi.org/10.1016/j.compedu.2016.09.005.

Mason, R. & Rennie, F. (2006). *Elearning: The Key Concepts*. Routledge.

Mayer, R. E. (2005). Principles of Multimedia Learning Based on Social Cues : Personalization, Voice, and Image Principles. In *The Cambridge Handbook of Multimedia Learning* (pp. 201–212). Cambridge University Press.

McCowen, G. (1998). An evaluation of the theoretical positions that underpin explanations of the nature of learning and learning difficulties and the relationship of these to literacy & literacy learning theories. *Australian Journal of Learning Disabilities*, *3*(3), 22–25, https://doi.org/10.1080/19404159809546567.

McInerney, D. M. & McInerney, V. (2006). *Educational psychology : constructing learning*. Frenchs Forest, N.S.W: Pearson Education Australia, 4th edition.

McLaren, B. M., DeLeeuw, K. E., & Mayer, R. E. (2011). Polite web-based intelligent tutors: Can they improve learning in classrooms? *Computers & Education*, *56*(3), 574–584, https://doi.org/10.1016/j.compedu.2010.09.019.

Meere, D. (2012). *An Infostation-Based Multi-Agent System Supporting Contextualised And Personalised Mlearning Service Delivery*. PhD thesis, University of Limerick, Ireland.

Mikic Fonte, F. A., Burguillo, J. C., & Nistal, M. L. (2012). An intelligent tutoring module controlled by BDI agents for an e-learning platform. *Expert Systems with Applications*, *39*(8), 7546–7554, https://doi.org/10.1016/j.eswa.2012.01.161.

Milton, M. (2000). How do schools provide for childrenwith learning difficulties in numeracy? *Australian Journal of Learning Disabilities*, *5*(2), 23–27, https://doi.org/10.1080/19404150009546623.

Moreno, R. (2005). Multimedia Learning with Animated Pedagogical Agents. In *The Cambridge Handbook of Multimedia Learning* (pp. 507–524). Cambridge University Press.

Moreno, R. & Flowerday, T. (2006). Students' choice of animated pedagogical agents in science learning: A test of the similarity-attraction hypothesis on gender and ethnicity. *Contemporary Educational Psychology*, *31*(2), 186–207, https://doi.org/10.1016/j.cedpsych.2005.05.002.

Moreno, R. & Mayer, R. E. (2000). A coherence effect in multimedia learning: The case for minimizing irrelevant sounds in the design of multimedia instructional messages. *Journal of Educational psychology*, *92*(1), 117.

Moriarty, B., Douglas, G., Punch, K., & Hattie, J. (1995). The importance of self-efficacy as a mediating variable between learning environments and achievement. *British Journal of Educational Psychology*, *65*(1), 73–84, https://doi.org/10.1111/j.2044-8279.1995.tb01132.x.

Mosa, M. J., Albatish, I., & Abu-Naser, S. S. (2018). ASP.NET-Tutor: Intelligent Tutoring System for leaning ASP.NET. *International Journal of Academic Pedagogical Research*, *2*(2), 1–8.

Mosavi, A., Bathla, Y., & Varkonyi-Koczy, A. (2018). Predicting the future using web knowledge: State of the art survey. In *Advances in Intelligent Systems and Computing*, volume 660 (pp. 341–349).: Springer Verlag.

Moulin, B. & Chaib-draa, B. (1996). An Overview of Distributed Artificial Intelligence. In G. O'Hare & N. Jennings (Eds.), *Foundations of Distributed Artificial Intelligence* chapter 1, (pp. 3–56). Wiley-interscience, 605 Third Avenue, New York, N.Y. 10158-0012.

Moundridou, M. & Virvou, M. (2002). Evaluating the persona effect of an interface agent in a tutoring system. *Journal of Computer Assisted Learning*, *18*(3), 253–261, https://doi.org/10.1046/j.0266-4909.2001.00237.x.

Muangprathub, J., Boonjing, V., & Chamnongthai, K. (2020). Learning recommendation with formal concept analysis for intelligent tutoring system. *Heliyon*, *6*(10), https://doi.org/10.1016/j.heliyon.2020.e05227.

Mubarak, A. A., Cao, H., & Zhang, W. (2020). Prediction of students' early dropout based on their interaction logs in online learning environment. *Interactive Learning Environments*, *0*(0), 1–20, https://doi.org/10.1080/10494820.2020.1727529.

Neal, J. C. & Kelly, P. R. (2002). Delivering the promise of academic success through late intervention. *Reading and Writing Quarterly*, *18*(2), 101–117, https://doi.org/10.1080/10573560252808503.

Nkambou, R., Bourdeau, J., & Mizoguchi, R. (2010). Introduction: What Are Intelligent Tutoring Systems, and Why This Book? In R. Nkambou, J. Bourdeau, & R. Mizoguchi (Eds.), *Advances in Intelligent Tutoring Systems*, volume 308 of *Studies in Computational Intelligence* (pp. 1–12). Berlin, Heidelberg: Springer Berlin Heidelberg.

Nwana, H. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, *4*(4), https://doi.org/10.1007/BF00168958.

Orey, M. A. & Nelson, W. A. (1993). Development principles for intelligent tutoring systems: Integrating cognitive theory into the development of computer-based instruction. *Educational Technology Research and Development*, *41*(1), 59–72, https://doi.org/10.1007/BF02297092.

Ormrod, J. E. (2005). *Educational psychology: Developing learners*. Upper Saddle River, NJ: Pearson-Merill, 5th edition.

O'Shea, T. (1979). *Self-Improving Teaching Systems*. Basel: Springer.

Paas, F., Renkl, A., & Sweller, J. (2004). Cognitive Load Theory: Instructional Implications of the Interaction between Information Structures and Cognitive Architecture. *Instructional Science*, *32*(1/2), 1–8, https://doi.org/10.1023/B:TRUC.0000021806.17516.d0.

Pammer-Schindler, V. (2020). From a Conversational Agent for Time Management towards a Mentor for (Study) Life Priorities: A Vision. In *CEUR Workshop Proceedings*, volume 3046.

Papanikolaou, K. A. (2015). Constructing Interpretative Views of Learners' Interaction Behavior in an Open Learner Model. *IEEE Transactions on Learning Technologies*, *8*(2), 201–214, https://doi.org/10.1109/TLT.2014.2363663.

Pham, Q. D. & Adina, M. F. (2013). Adaptation to Learners' Learning Styles in a Multi-Agent E-Learning System. *Internet Learning*, 2, https://doi.org/10.18278/il.2.1.2.

Pierrakeas, C., Koutsonikos, G., Lipitakis, A.-D., Kotsiantis, S., Xenos, M., & Gravvanis, G. A. (2020). The Variability of the Reasons for Student Dropout in Distance Learning and the Prediction of Dropout-Prone Students. In M. Virvou, E. Alepis, G. A. Tsihrintzis, & L. C. Jain (Eds.), *Machine Learning Paradigms: Advances in Learning Analytics* chapter 6, (pp. 91–111). Cham: Springer International Publishing, first edition.

Pistilli, M. D. & Arnold, K. E. (2010). In practice: Purdue Signals: Mining real-time academic data to enhance student success. *About Campus*, 15(3), 22–24, https://doi.org/10.1002/abc.20025.

Pressey, S. L. (1926). A simple apparatus which gives tests and scores - and teaches. *School and Society*, 23(586), 373–376.

Rajabalee, B. Y., Santally, M. I., & Rennie, F. (2020). A study of the relationship between students' engagement and their academic performances in an eLearning environment. *E-Learning and Digital Media*, 17(1), 1–20, https://doi.org/10.1177/2042753019882567.

Ramos, C., Frasson, C., & Ramachandran, S. (2009). Introduction to the Special Issue on Real World Applications of Intelligent Tutoring Systems. *IEEE Transactions on Learning Technologies*, 2(2), 62–63, https://doi.org/10.1109/TLT.2009.26.

Razali, S. N. A. M., Rusiman, M. S., Gan, W. S., & Arbin, N. (2018). The Impact of Time Management on Students' Academic Achievement. *Journal of Physics: Conference Series*, 995(2018), 12–42, https://doi.org/10.1088/1742-6596/995/1/012042.

Reyes, R., Garza, D., Garrido, L., De la Cueva, V., & Ramirez, J. (2019). Methodology for the Implementation of Virtual Assistants for Education Using Google Dialogflow. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11835 LNAI (pp. 440–451). Springer.

Rivera, A. C., Tapia-Leon, M., & Lujan-Mora, S. (2018). Recommendation Systems in Education: A Systematic Mapping Study. In *Advances in Intelligent Systems and Computing*, volume 721 (pp. 937–947). Springer Verlag.

Roberts, F. C. (1984). An overview of intelligent CAI systems. *Peabody Journal of Education*, 62(1), 40–51, https://doi.org/10.1080/01619568409538463.

Robertson, P., Hamill, P., & Hewitt, C. (1994). *Effective support for learning*. Technical report, Education Department Scottish Office, Edinburgh, UK.

Robinson, G. (2002). Assessment of learning disabilities: The complexity of causes and consequences. *Australian Journal of Learning Disabilities*, 7(1), 29–39, https://doi.org/10.1080/19404150209546690.

Rogers, E. M. (1995). *Diffusion of innovations*. New York: Free Press, 4th ed. edition.

Rogers, P. L. (2001). Traditions to Transformations: The Forced Evolution of Higher Education. *AACE Journal*, 9(1), 47–60.

Rosenberg-Kima, R. B., Baylor, A. L., Plant, E. A., & Doerr, C. E. (2007). The Importance of Interface Agent Visual Presence: Voice Alone Is Less Effective in Impacting Young Women's Attitudes Toward Engineering. In Y. de Kort, W. IJsselsteijn, C. Midden, B. Eggen, & B. J. Fogg (Eds.), *Persuasive Technology*, volume 4744 of *Lecture Notes in Computer Science* (pp. 214–222). Berlin, Heidelberg: Springer Berlin Heidelberg.

Rosner, J. (1994). Helping Children Overcome Learning Difficulties. *Optometry and Vision Science*, *71*(7), 474–475, https://doi.org/10.1097/00006324-199407000-00008.

Russell, S. J. & Norvig, P. (1995). Intelligent Agents. In *Artificial Intelligence: A Modern Approach* chapter 2, (pp. 31–52). NJ: Prentice Hall, Upper Saddle River, 1st edition.

Russell, S. J. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. NJ: Prentice Hall, Upper Saddle River, 3rd edition.

Sääksjärvi, M. & Hellén, K. (2019). Idea selection using innovators and early adopters. *European Journal of Innovation Management*, *22*(4), 585–599, https://doi.org/10.1108/EJIM-05-2018-0094.

Saleh, S., Qader, A. N., Zeebaree, M., Ismael, G. Y., & Aqel, M. (2021). Time management during corona virus era. *Estudios de Economia Aplicada*, *39*(7), https://doi.org/10.25115/eea.v39i7.5233.

Sànchez-Marrè, M. (2022). *Intelligent Decision Support Systems*. Cham: Springer International Publishing.

Sandoval, A., Gonzalez, C., Alarcon, R., Pichara, K., & Montenegro, M. (2018). Centralized student performance prediction in large courses based on low-cost variables in an institutional context. *The Internet and Higher Education*, *37*, 76–89, https://doi.org/10.1016/j.iheduc.2018.02.002.

Santrock, J. W. (2006). *Life-span development*. Boston, MA: Mcgraw Hill, 5th edition.

Sarra, A., Fontanella, L., & Di Zio, S. (2019). Identifying Students at Risk of Academic Failure Within the Educational Data Mining Framework. *Social Indicators Research*, *146*(1-2), 41–60, https://doi.org/10.1007/s11205-018-1901-8.

Saykılı, A. (2018). Distance education: Definitions, generations, key concepts and future directions. *International Journal of Contemporary Educational Research*, *5*(1), 2–17.

Schroeder, J. I., et al. (2013). Using membrane transporters to improve crops for sustainable food production. *Nature*, *497*(7447), 60–66, https://doi.org/10.1038/nature11909.

Schroeder, N. L. & Adesope, O. O. (2014). A Systematic Review of Pedagogical Agents' Persona, Motivation, and Cognitive Load Implications for Learners. *Journal of Research on Technology in Education*, *46*(3), 229–251, https://doi.org/10.1080/15391523.2014.888265.

Schwabe, A. (2021). JARET: A Human Assistive A.I. Agent for Goal Review and Time Management. *Indonesian Journal Of Educational Research and Review*, *4*(3), 392, https://doi.org/10.23887/ijerr.v4i3.41630.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences*, *3*(3), 417–424, https://doi.org/10.1017/S0140525X00005756.

Seligman, M. E. P. (1995). *The optimistic child: How learned optimism protects children from depression*. Boston, MA: Houghton Mifflin.

Shahabadi, M. M. & Uplane, M. (2015). Synchronous and Asynchronous e-learning Styles and Academic Performance of e-learners. *Procedia - Social and Behavioral Sciences*, *176*, 129–138, https://doi.org/10.1016/j.sbspro.2015.01.453.

Shute, V. & Psotka, J. (1994). *Intelligent Tutoring Systems: Past, Present, and Future*. Technical report, Brooks Air Force Base, Texas.

Siegel, L. & Brayne, R. (2005). *Early Identification and Intervention to Prevent Reading Difficulties*. Technical report, CMEC Toronto.

Simonson, M., Zvacek, S. M., & Smaldino, S. (2019). *Teaching and Learning at a Distance: Foundations of Distance Education*. Information Age Publishing, 7th edition.

Slavin, R., Karweit, N., & Wasik, B. (1994). *Preventing early school failure: research, policy, and practice*. Boston: Allyn & Bacon., 1st edition.

Sleeman, D. & Brown, J. (1982). Introduction: Intelligent tutoring systems. In D. Sleeman & J. Brown (Eds.), *Intelligent tutoring systems*. London: Academic Press.

Subagdja, B., Sonenberg, L., & Rahwan, I. (2009). Intentional learning agent architecture. *Autonomous Agents and Multi-Agent Systems*, *18*(3), 417–470, https://doi.org/10.1007/s10458-008-9066-5.

Sugai, G. & Evans, D. (1997). Using teacher perceptions to screen for primary students with high risk behaviours. *Australasian Journal of Special Education*, *21*(1), 18–35, https://doi.org/10.1080/1030011970210104.

Sun, S., Joy, M. S., & Griffiths, N. E. (2007). The use of learning objects and learning styles in a multi-agent education system. *Journal of Interactive Learning Research*, *18*(3), 381–398.

Sutton, R. S. & Barto, A. G. (2012). *An Introduction to Reinforcement Learning*. Cambridge, Massachusetts: MIT Press, 2nd edition.

Sycara, K. P. (1998). Multiagent systems. *AI Magazine*, *19*(2), 14.

Tarimo, W., Deeb, F., & Hickey, T. (2016). Early detection of at-risk students in CS1 using teachback/spinoza. *Journal of Computing Sciences in Colleges*, *31*(6), 105–111.

Terzieva, V., Ilchev, S., Todorova, K., & Andreev, R. (2021). Towards a Design of an Intelligent Educational System. *IFAC-PapersOnLine*, *54*(13), 363–368, https://doi.org/10.1016/j.ifacol.2021.10.474.

Tkalcic, M. & Chen, L. (2015). Personality and Recommender Systems. In *Recommender Systems Handbook* (pp. 715–739). Boston, MA: Springer US.

Tkalčič, M. & Chen, L. (2022). Personality and Recommender Systems. In *Recommender Systems Handbook* (pp. 757–787). New York, NY: Springer US.

Toti, D., Capuano, N., Campos, F., Dantas, M., Neves, F., & Caballé, S. (2021). Detection of Student Engagement in e-Learning Systems Based on Semantic Analysis and Machine Learning. *Lecture Notes in Networks and Systems*, *158 LNNS*, 211–223, https://doi.org/10.1007/978-3-030-61105-7_21.

Truong, H. M. (2016). Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities. *Computers in Human Behavior*, *55*, 1185–1193, https://doi.org/10.1016/j.chb.2015.02.014.

Tsai, K.-P., Yang, F.-C., & Tang, C.-Y. (2022). Multiagent mobility and lifestyle recommender system for individuals with visual impairment. *Neuroscience Informatics*, *2*(4), 100077, https://doi.org/10.1016/j.neuri.2022.100077.

Turing, A. M. (1950). I.—Computing Machinery And Intelligence. *Mind*, *59*(236), 433–460, https://doi.org/10.1093/mind/LIX.236.433.

Twaddell, P. (2001). If you want to make a difference, intervene: He Austrlian kindergarten screening instrument. *Australian Journal of Learning Disabilities*, *6*(4), 26–35, https://doi.org/10.1080/19404150109546684.

Twomey, E. (2006). Linking learning theories and learning difficulties. *Australian Journal of Learning Disabilities*, *11*(2), 93–98, https://doi.org/10.1080/19404150609546812.

Valas, H. (2001). Learned helplessness and psychological adjustment II: Effects of learning disabilities and low achievement. *Scandinavian Journal of Educational Research*, *45*(2), 101–114, https://doi.org/10.1080/00313830120052705.

Van Bragt, C. A. C., Bakx, A. W. E. A., Bergen, T. C. M., & Croon, M. A. (2011). Looking for students' personal characteristics predicting study outcome. *Higher Education*, *61*(1), 59–75, https://doi.org/10.1007/s10734-010-9325-7.

Van Kraayenoord, C. & Elkins, J. (1998). Learning difficulties in regular classrooms. In *Educating children with special needsLtd*. Sydney, Australia: Prentice Hall.

VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, *46*(4), 197–221, https://doi.org/10.1080/00461520.2011.611369.

Vanlehn, K., et al. (2005). The Andes Physics Tutoring System: Five Years of Evaluations. In *International Journal of Artificial Intelligence in Education*, volume 15 (pp. 678–685).

Veletsianos, G. & Russell, G. S. (2014). Pedagogical Agents. In *Handbook of Research on Educational Communications and Technology* (pp. 759–769). New York, NY: Springer New York.

Virginia, D. & Julian, P. (2013). Multiagent Organizations. In G. Weiss (Ed.), *Multi-Agent Systems* chapter 2, (pp. 51–98). Cambridge, Massachusetts: The MIT Press, 2nd edition.

Von Hippel, P. T. & Hofflinger, A. (2021). The data revolution comes to higher education: identifying students at risk of dropout in Chile. *Journal of Higher Education Policy and Management*, *43*(1), 2–23, https://doi.org/10.1080/1360080X.2020.1739800.

Waddington, R. J., Nam, S., Lonn, S., & Teasley, S. D. (2016). Improving Early Warning Systems with Categorized Course Resource Usage. *Journal of Learning Analytics*, *3*(3), 263–290, https://doi.org/10.18608/jla.2016.33.13.

Wan, S. & Niu, Z. (2018). An e-learning recommendation approach based on the self-organization of learning resource. *Knowledge-Based Systems*, *160*, 71–87, https://doi.org/10.1016/j.knosys.2018.06.014.

Wang, N., Johnson, W. L., Mayer, R. E., Rizzo, P., Shaw, E., & Collins, H. (2008). The politeness effect: Pedagogical agents and learning outcomes. *International Journal of Human-Computer Studies*, *66*(2), 98–112, https://doi.org/10.1016/j.ijhcs.2007.09.003.

Wasik, B. A., & Karweit, N. L. (1994). Off to a good start: Effects of birth to three interventions on early school success. In R. E. Slavin, N. L. Karweit, & B. A. Wasik (Eds.), *Preventing early school failure*. Boston: Allyn & Bacon.

Wearmouth, J. (2002). The Role of the Learning Support Co-ordinator: Addressing the Challenges. In *Dyslexia and Literacy: Theory and Practice* (pp. 213–228). Chichester: Wiley.

Weil, A. R. (2018). Diffusion Of Innovation. *Health Affairs*, *37*(2), 175–175, https://doi.org/10.1377/hlthaff.2018.0059.

Weiss, G. (1999). Prolog. In G. Weiss (Ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence* (pp. 644). MIT Press.

Westwood, P. (1995). Teachers' beliefs and expectations concerning students with learning difficulties. *Australian Journal of Remedial Education*, *27*(2), 19–21.

Westwood, P. S. (2004). *Learning and Learning Difficulties: A handbook for teachers*. ACER Press.

Westwood, P. S. (2006). *Teaching and Learning Difficulties: Cross-curricular perspectives*. Victoria, Australia: ACER Press, 19 Prospect Hill Road, Camberwell Victoria, 3124, Australia.

Westwood, P. S. (2008). *What teachers need to know about Learning difficulties*. Victoria, Australia: ACER Press, 19 Prospect Hill Road, Camberwell Victoria, 3124, Australia.

Westwood, P. S. (2013). *Reading and Learning Difficulties: Approaches to teaching and assessment*. ACER Press.

Willging, P. A. & Johnson, S. D. (2004). Factors That Influence Students' Decision to Dropout of Online Courses. *Online Learning*, *8*(4), 115–127, https://doi.org/10.24059/olj.v8i4.1814.

Wilson, V. (2006). *Does Small Really Make a Difference ? An update: A review of the literature on the effects of class size on teaching practice and pupils' behaviour and attainment*. Technical Report May, The Scottish Council for Research in Education, Edinburgh.

Wolff, A., Zdrahal, Z., Herrmannova, D., Kuzilek, J., & Hlosta, M. (2014). Developing predictive models for early detection of at-risk students on distance learning modules. In *The 4th International Conference on Learning Analytics and Knowledge (LAK14)*, volume 1137 (pp.$\tilde{6}$).

Wooldridge, M. (1999). Intelligent Agents. In G. Weiss (Ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence* chapter 1, (pp. 27–78). The MIT Press Cambridge, Massachusetts.

Wooldridge, M. (2013). Intelligent Agents. In G. Weiss (Ed.), *Multi-Agent Systems* chapter 1, (pp. 3–50). Cambridge, Massachusetts: The MIT Press, 2nd edition.

Wooldridge, M. & Jennings, N. R. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, *10*(2), 115–152, https://doi.org/10.1017/S0269888900008122.

Woolf, B. P. (2008). *Building Intelligent Interactive Tutors*. Elsevier, 1st edition.

Wright, R. J. (2003). A mathematics recovery: Program of intervention in early number learning. *Australian Journal of Learning Disabilities*, *8*(4), 6–11, https://doi.org/10.1080/19404150309546741.

You, J. W. (2016). Identifying significant indicators using LMS data to predict course achievement in online learning. *The Internet and Higher Education*, *29*, 23–30, https://doi.org/10.1016/j.iheduc.2015.11.003.

Yukselturk, E., Ozekes, S., & Türel, Y. K. (2014). Predicting Dropout Student: An Application of Data Mining Methods in an Online Education Program. *European Journal of Open, Distance and E-Learning*, *17*(1), 118–133, https://doi.org/10.2478/eurodl-2014-0008.

Zakharov, K. & Mitroví, A. (2004). *Feedback micro-engineering in EER-Tutor*. Technical report, University of Canterbury.

Zeiser, K., Scholz, C., & Cirks, V. (2018). *Maximizing Student Agency: Implementing and Measuring Student-Centered Learning Practices.* Technical Report October, American Institutes for Reasearch, Whashinton, DC.

Zhang, L.-f. (2007). Do personality traits make a difference in teaching styles among Chinese high school teachers? *Personality and Individual Differences*, *43*(4), 669–679, https://doi.org/10.1016/j.paid.2007.01.009.

Zhang, X., Xu, H., Hu, L., & Zhuang, S. (2011a). Collaborative Learning on Multi-agent in M-Learning. In K. Arai (Ed.), *Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 1*, Lecture Notes in Networks and Systems (pp. 595–601). Cham: Springer International Publishing.

Zhang, X., Xu, H., Hu, L., & Zhuang, S. (2011b). Collaborative Learning on Multi-agent in M-Learning. In *Advances in Intelligent and Soft Computing* (pp. 595–601). Cham: Springer.

Zhou, Y., Zhao, J., & Zhang, J. (2020). Prediction of learners' dropout in E-learning based on the unusual behaviors. *Interactive Learning Environments*, (pp. 1–25)., https://doi.org/10.1080/10494820.2020.1857788.

Zigerell, J. (1984). *Distance education: An information age approach to Adult Education*. Technical report, ERIC Clearinghouse on Adult, Career, and Vocational Education, Columbus, Ohio, USA.

Appendices

Appendix A

Existing Multi-Agent System Platforms

A.1 JADE (Java Agent DEvelopment Framework)

JADE[1] is a popular software platform for creating MAS applications. The platform consists of three essential parts accommodating to the FIPA (Foundation for Intelligent Physical Agents) standards.

- **The Agent Management System (AMS)** regulates agent registration and authentication, maintains agent life cycle, and offers white page services (the register of existing agents).
- **The Director Facilitator (DF) module** offers yellow page services (the agent services registered in the AMS).
- **The Agent Communication Channel (ACC) module** controls inter-agent communication.

In JADE, agents use the asynchronous messaging paradigm to communicate according to the built-in FIPA ACL (Agent Communication Language) language. Each message comprises the sender, recipient, communication, and content fields. The message elements can be a digital or logical expression (primitive) or a user-defined structure (aggregated of primitives), usually embedded as Java, XML, or bytecode classes. JADE-based multi-agent applications are widely applied in industry, education, and science.

[1] JADE. *https://jade.tilab.com/* (Accessed: 14, March 2022)

A.2 MadKIT (Multi-Agent Development KIT)

MadKIT[2] is a modular and scalable multi-agent platform built on the Agent/Group/Role (AGR) concept with initial open code developed in Java. The platform enables networked application development and multi-agent paradigm modeling. One of the most critical features of MADKit is that it employs the OCMAS (Organization-centered Multi-AgentSystem) approach, which differs from traditional techniques that are primarily agent-oriented. The following are the significant aspects of MaDKit:

- it allows the construction of intelligent agents and lifecycle control,
- the establishment of an infrastructure for providing agent communication and structuring multi-agent models, and
- facilitates the development of distributed agent-based applications.

Most multi-agent modeling applications for the industry are built using MADKit.

A.3 JACK Intelligent Agents

"JACK Intelligent Agents[3]" is a cross-platform development framework for developing and integrating commercial and industrial systems written in Java. It is built on the experiences of the Procedural Reasoning System (PRS) and Distributed multi-agent Reasoning System (DMARS). JACK is one of the few multi-agent systems to employ the BDI (Beliefs/Desires/Intentions) software paradigm, and it comes with its Java-based plan language and graphical planning tools. Each agent is assigned a role in the environment based on its design objectives, knowledge, and communication skills and then performs it autonomously.

JACK is a powerful cross-platform due to these features:

- its minimal computational resource needs, suited for processing hundreds of agents on low-level hardware,
- its "Transparent" agent communication, and
- its availability as a graphical tool for building agents.

A.4 Agent Builder

"Agent Builder[4]" is a suite of tools for constructing collaborating intelligent software agents. The Agent Builder Toolkit (ABT) and the compilation environment are

[2] MadKIT. *http://www.madkit.net/madkit/madkit.php* (Accessed: 14, March 2022)

[3] JACK. *https://aosgrp.com/products/jack/* (Accessed: 14, March 2022)

[4] Agent Builder. *https://www.agentbuilder.com/Documentation/ReferenceManual-v1.4.pdf* (Accessed: 15, March 2022)

the two primary components of the Agent Builder. The ABT includes tools for managing multi-agent software development, analyzing agent transactions, designing and developing interconnected agents, determining individual agent behavior, and debugging and testing agents. Agents created by the ABT use the Knowledge Management Language (KML) and the Knowledge Query and Manipulation Language (KQML) requests to communicate and support a variety of information exchange functions. The ABT is a commercial tool for creating intelligent agents and multi-agent applications rapidly and simply. It features flexible project management capabilities, allowing MAS developers to manage software development successfully.

– **The Project Manager** is a high-level tool for developing projects, agencies, and agents. The Agency Manager is used to construct agencies, and the Agent Manager is used to create agents. The Project Manager gives a high-level overview of the development process. The Project Manager makes it simple to see all the projects, agencies, and agents you have established.

– **The Ontology Manager** helps the developer analyze the agent's issue area and determine the vital structure of the agent's work.

– **The Agency Manager** displays the activity of the whole agent system in a temporary user window. As a result, the developer can keep track of inter-agent communications and use a debugger to check individual agents' conditions.

– **The Agent Manager** offers a tool for customizing an individual agent's behavior. This program comes with a graphical editor for configuring an agent's essential characteristics, such as beliefs, commitments, intents, opportunities, and behavior. The agent manager also has the capability of planning and training the agent.

– **The Protocol Manager** allows building and displaying a set of protocols to use with different agencies. This program provides a high-level overview of protocols. The Protocol Editor is where all attributes required to utilize a protocol with a particular agency are defined.

A.5 Cougaar (Cognitive Agent Architecture)

Cougaar[5] is a Java-based architecture that enables constructing large-scale distributed agent-based applications that focus on optimizing logistics support to operations and revolutionizing global military logistics by improving coordination between operations and logistics to enhance overall operational performance. It is built over research conducted by the Defense Advanced Research Projects Agency (DARPA) to determine the feasibility of using advanced agent technology to perform rapid, large-scale, distributed logistics planning and replanning. Cougaar agents have a framework that comprises numerous integrated extended services:

- a "Blackboard" for agent communication;
- HTTP services (HyperText Transfer Protocol) for user interfaces;

[5] Cougaar. *http://www.cougaarsoftware.com/intelligent-systems/our-technology-approach/* (Accessed: 15, March 2022)

- knowledge representation systems;
- coordination amongst agents via assignment mechanisms.

A.6 Zeus Agent Building Toolkit

Zeus[6] is a toolset for building collaborative multi-agent applications. It is an integrated environment enabling the quick creation of multi-agent systems based on known agent technologies. ZEUS provides a visual environment to capture user specifications for agents used to produce Java source code. ZEUS aimed to speed up new multi-agent-based applications development by abstracting multi-agent systems' common concepts and components. The objective was to provide a collaborative agent-building toolset that software engineers with minimal agent technology knowledge could utilize to construct viable multi-agent systems. Thus, our design philosophy embodied the following:

It should first distinguish between domain-level issue solving and agent functioning. The former covers the assimilation, representation, and use of domain-specific knowledge in problem resolution. With the offered agent-level functionality, the developers could focus on developing their agents' domain-specific problem-solving skills.

Second, the toolkit should be used in a Visual Programming style. Therefore, the toolkit would facilitate agents' building by offering organized menus and tables that would allow application developers to customize agent capabilities and modalities easily.

Third, the toolkit should be open-source and easily extendable. To set up additional agents, skilled users may easily add user-defined components to the library of system-supplied components.

A.7 MASON

MASON[7] is a fast discrete-event multiagent simulation library core in Java intended to serve as the foundation for extensive custom-purpose Java simulations while also providing more than enough capability for many lightweight simulation purposes. It includes a model library and an optional 2D and 3D visualization tools suite. It is a high-performance Java framework for multi-agent modeling of diverse processes, including graphical representation options in two- and three-dimensional formats. Mason is a powerful tool capable of quick data processing, low computer resource requirements, and its ability to integrate with other applications; most importantly, it can display 2D or 3D visualization of the chosen agents. The MASON platform creates practical applications for various activities, such as urban traffic modeling, aerial surveillance, and armed drones. Many works relied on Mason to create simulation models for Politics, Environment, Insurgency, Sustainable Development, Climate Change, and Conflict among Pastoralists in East Africa.

[6] Zeus3 *http://zeusagent.sourceforge.net/docs/techmanual/part1.html* (Accessed: 17, March 2022)

[7] MASON. *https://cs.gmu.edu/ eclab/projects/mason/* (Accessed: 17, March 2022)

A.8 Other MAS Platforms

In the following is presented an exhaustive list of other MAS Platforms.

- **SemanticAgent** is based on JADE and allows the development of agents whose behavior is represented in SWRL. SemanticAgent is developed at LIRIS; it is open-source and licensed under GPL V3.
- **Janus** is a modular multi-agent platform written in Java. It allows the creation of MAS with or without an organizational approach based on the Capacity-Role-Interaction-Organization (CRIO) model. Janus also proposes a system simulation model assimilating agents to holons (or recursive agents). Janus is extensible thanks to its pen Services Gateway initiative (OSGi) modules and offers network support via the JXTA library. A methodology named ASPECS can be associated with Janus. This platform is jointly developed by (en) ICAP-SeT-UTBM in France and (es) CITAT in Argentina.
- **GAMA** is an open-source simulation platform (under LGPL license) offering a spatially explicit agent-based modeling environment (using GIS data to describe the agents and their environment). IRD/UPMC developed them within the international joint unit UMMISCO.
- **CORMAS (COmmon Resources Multi-Agent System)** is an open-source MAS development framework based on the Smalltalk object-oriented programming language. It is a spatialized system focused on research problems in development sciences and negotiation between actors.
- **DoMIS** is a tool for the design of Multi-Agent Systems (oriented "operational management of complex systems") and based on the B-ADSC design method focused on design, DoMIS allows the establishment of specifications usable by any development platform capable of simulating, at best, real-time.
- **Jadex** is an agent platform developed in JAVA by the University of Hamburg that is modular, compatible with many standards, and capable of developing agents following the BDI model.
- **JAgent** is an open-source framework made in Java whose objective is to facilitate the development and testing of multi-agent systems.
- **Jason** is an open-source agent development environment in the AgentSpeak21 formalism developed in Java by Jomi Fred Hübner and Rafael H. Bordini.
- **MAGIQUE** is a platform for physically distributed agents written in Java and providing an original call-to-action communication model. In MAGIQUE, skills are dissociated from agents. The architecture of the agents and the different skills are developed separately. The skills are then grafted as plugins into the agents at the designer's discretion. This platform is developed within the LIFL.
- **OMAS (Open Multi-Agent Asynchronous Systems)** is a research platform developed by the artificial intelligence team of the University of Technology of Compiègne, under the direction of Jean-Paul Barthès.
- **SPADE** is a development environment for multi-agent organizations based on the XMPP protocol and is written in Python.

– **MASSIVE** is a multi-agent-based crowd simulation software that has enabled the creation of special effects in a large number of movies, having been originally developed for the fight scenes in The Lord of the Rings.

– **Golaem Crowd** is a multi-agent-based plugin for Maya software that allows crowd simulation for special effects directly in Maya.

– **Simulate** a real-time 3D multi-agent traffic simulation platform developed by Voxelia in partnership with CITAT and ICAP-SeT-UTBM.

– **NetBioDyn** is an easy-to-use multi-agent simulation tool for education developed at the University of Bretagne Occidentale.

– **AnyLogic** - Multi-agent and multi-method simulation software

Appendix B

Types of Agents' Environment

According to Russell and Norvig (2010), There are several types of agent environments. The following are the main distinctions to be made:

B.1 Accessible vs. inaccessible

If an agent's sensory system permits it to view the whole state of the environment, we say the environment is accessible to it. The environment is successfully accessible if the sensors detect all aspects significant to the action's choice. An accessible environment is helpful because it eliminates the need for the agent to keep track of the world without maintaining any internal state.

B.2 Deterministic vs. non-deterministic

An environment is deterministic when both its current state alongside the agents' actions can ultimately dictate the future state of the environment. In a predictable, accessible world, an agent does not need to be concerned about uncertainty. However, if the environment is unavailable, it may look non-deterministic; this is especially true if the environment is complicated, making it difficult to keep track of all the inaccessible features. As a result, from the agent's perspective, it is generally better to conceive of an environment as deterministic or non-deterministic.

B.3 Episodic vs. non-episodic

The agent's experience in an episodic environment is separated into "episodes." Each episode follows the agent's perception and subsequent action. The episode in question solely determines the quality of the agent's action while the following episodes are unaffected by prior episodes' activities. An episodic environment is straightforward because the agent does not have to prepare plans.

B.4 Static vs. dynamic

When an agent's environment may change while deliberating, we call it dynamic; otherwise, we call it static. Static settings are easier to manage, as the agent does not need to keep looking out the window to choose what to do, nor does it need to worry about the time elapsing. When the environment does not vary over time, but the agent's performance score does, the environment is said to be semi-dynamic.

B.5 Discrete vs. continuous

The environment is discrete if there are a small number of distinct, clearly defined percepts and actions. Chess is discrete in that each turn has a set number of possible moves. The speed and position of the cab and other cars sweep over a range of continuous values when operating a taxi.

We listed the qualities of various recognizable environments in Table B.1. It is important to note that the agent's actions may vary according to its environment's settings. A poker game, for example, is deterministic if the agent can keep track of the deck's order but non-deterministic if it can't. At a higher level than the agent's actions, many settings are episodic; A chess tournament, for example, consists of a series of games; each game is an episode since (for the most part), the contribution of moves in one game to the agent's total performance is unaffected by movements in the next game. Steps inside a single game, on the other hand, almost always interact; therefore, the agent must plan multiple early moves (Russell and Norvig, 2010).

Environment	Accessible	Deterministic	Episodic	Static	Discrete
Chess with a clock	Yes	Yes	No	Semi	Yes
Chess without a clock	Yes	Yes	No	Yes	Yes
Poker game	No	No	No	Yes	Yes
Backgammon game	Yes	No	No	Yes	Yes
Taxi driving	No	No	No	No	No
Medical diagnosis system	No	No	No	No	No
Image-analysis system	Yes	Yes	Yes	Semi	No
Part-picking robot	No	No	Yes	No	No
Interactive English tutor	No	No	No	No	Yes

Table B.1 – *Examples of environments and their characteristics.*

Given enough complexity, determinism becomes irrelevant, and an environment may as well be non-deterministic than deterministic. Inaccessible, non-deterministic, non-episodic, dynamic, and continuous environments are the most complicated (Russell and Norvig, 2010).