



HAL
open science

Intelligent Tutoring Systems for Psychomotor Development in Open Environments

Laurentiu Neagu

► **To cite this version:**

Laurentiu Neagu. Intelligent Tutoring Systems for Psychomotor Development in Open Environments. Technology for Human Learning. Université Paris sciences et lettres; Universitatea politehnica (Bucarest), 2022. English. NNT : 2022UPSLM103 . tel-04751910

HAL Id: tel-04751910

<https://pastel.hal.science/tel-04751910v1>

Submitted on 24 Oct 2024

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

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



THÈSE DE DOCTORAT DE L'UNIVERSITÉ PSL

Préparée à Mines Paris-PSL

Dans le cadre d'une cotutelle avec l'Université Polytechnique de Bucarest

Un système de tutorat intelligent pour le développement des compétences psychomotrices dans des environnements ouverts

Intelligent Tutoring Systems for Psychomotor Development in Open Environments

Soutenue par

Laurentiu-Marian NEAGU

Le 21 novembre 2022

Ecole doctorale n° 621

Ingénierie des Systèmes, Matériaux,
Mécanique, Énergétique (ISMME)

Sciences et Génie des Activités à Risques

Composition du jury :

Mihai, DASCALU
Professeur, University
Politehnica of Bucharest *Président*

Elvira, POPESCU
Professeur,
University of Craiova *Rapporteur*

Monica, STANESCU
Professeur, National university
of physical training and sport *Rapporteur*

Liviu P., DINU
Professeur,
University of Bucharest *Examineur*

Andon TCECHMEDJIEV
Maitre de conférences IMT ALES *Examineur*

Sébastien, TRAVADEL
Directeur de recherche,
MINES Paris *Directeur de thèse*

Răzvan RUGHINIS,
Professeur, University
Politehnica of Bucharest *Directeur de thèse*

Acknowledgments

This thesis is the result of the close collaboration between two research groups – MINES ParisTech Centre of Research in Risks and Crisis, from Sophia Antipolis, France, and the Human-Computer Interaction Research Centre in Politehnica University of Bucharest, Romania. The people who contributed to the current research were many, and the results obtained were thanks to the close collaboration and teamwork, in various contexts, over the past years.

Foremost, in the center of this network are my mentors and supervisors, *Sébastien Travadel* and *Răzvan Rughiniș*, to whom I am extremely grateful for their guidance, trust, and their continuous adjustments along the way - to see the overall picture when I was sometimes lost in details.

Afterward comes a close circle of collaborators who provided either guidance or direct involvement in this project. I will start with *Mihai Dascălu*, a full professor at the Politehnica University of Bucharest, who has also been a close friend throughout these years. I consider Mihai the person from which I learned how to frame experiments and have the critical thinking of a researcher. Also, I want to thank *Eric Rigaud* – researcher at MINES ParisTech, for his continuous guidance and his infinite patience in revising my work. Eric was the person with whom I spent the most time while working on the thesis, and he was the one with whom I designed the first rules of training – the athletisation program, and further developed together *OntoStrength*. Next, I want to thank *Vincent Guarnieri* – a sports scientist at MINES ParisTech, for his support in building the knowledge base of exercises, links with movements, muscles, joints, and their full description. Also, together with Eric, they helped me on designing the training rules, calibration protocol, and personalization parameters.

The following level, although the boundaries fade, consists firstly of an engineer from the Politehnica University of Bucharest, *Emanuel Radu*, who helped me in refining *OntoStrength* and developing an application interface to integrate the ontology with external systems. Together with him, I had the pleasure to collaborate with *Ștefan Chelbosu* and *Andrei Pancu*, Computer Science engineers, who helped me with the development of *Selfit* user interface assessment modules and third-party integrations. Throughout my journey, I also had the support of *Didier Delaitre*, medic, and researcher at MINES ParisTech, who helped me on developing the injury risk assessment module in *Selfit*.

I am grateful for *Franck Guarnieri's* openness and kindness throughout these years, and his stubbornness to improve my French. Last, but under no circumstances least, I am extremely grateful to my wife, my parents, and my friends for all their support and understanding

Abstract

Information technology has produced incremental changes in nearly all industries, whereas recent research on technology applied in education has shown the potential to generate an inflection point in the way people learn. The current educational revolution is mainly driven by three components: Artificial Intelligence, Internet availability, and advancements in educational sciences. Intelligent Tutoring Systems (ITSs) are considered one of the best-fitted candidates for educational transformation. An ITS is a computer-based system that produces personalized tutoring through individualized, pedagogically sound, and easy-to-access educational material. ITSs engage students independently or collaboratively to ensure effective learning. The AI in Education (AIED) research groups explored various methods and assumptions for building efficient tutors in the cognitive field, with notable results in disciplines like physics, mathematics, and informatics. In contrast, the psychomotor field is exhibiting only lately an intensive digitalization process, as more recent approaches are introducing intelligent tutors for this field.

The main goal of this thesis is to provide personalized sports training sessions in the psychomotor field in the form of an ITS – *Selfit* – an efficient and easy-to-use system that has the long-term goal to engage people in sports and improve the general health of the mass population.

This objective is three-fold. First, we introduce an ontology to model key concepts of the psychomotor field – representations, and relations between the concepts, data, and entities. The ontology for knowledge modeling in *Selfit*, called *OntoStrength*, was built using the Ontology Development 101 framework and employing a multi-disciplinary team, with sports, medical, and computer science specialists.

Second, we introduce a contextual multi-armed bandit algorithm for generating personalized training sessions. The decision-making process of a psychomotor tutor proves difficult. There are many unknown variables and uncertainty: the training time is limited, the trainee cannot test all the activities, and the personalization should happen in real time while maintaining the user motivated and engaged. The *Selfit* approach for psychomotor tutoring has proven to surpass the fixed-rules training approach in our simulations.

Third, we evaluate the utility and effectiveness of our ITS prototype on a population of 42 users, with low and medium training experience, which were involved in an experiment, that included two adaptive strategies for tutoring – one narrow, and the other with a wide exploration space. *Selfit* has a user-friendly mobile interface, where the user can visualize the video training content to execute and is required to assess the effort implied by each training component. A usability and experience survey was filled out at the end of the experiment. The users generally perceived *Selfit* as practical,

predictable, simple, connective, stylish, motivating, novel, and captivating. The results are also in line with our initial simulations, proving the potential of the proposed approach in personalized training.

Selfit evaluation showed promising results and highlighted the usefulness of the ITS architecture in the psychomotor field. The current thesis can be considered the foundation of a new crossroad, between AIED and psychomotor training, opening new research directions aiming to improve the general health of the population through automated systems.

Future work aims to extend the knowledge base from strength training to other training types, such as flexibility and mobility, enrich the user experience by providing voice support while training, as well as integrate NLP techniques to enhance tutor-trainee interaction and computer vision algorithms for real-time assessment.

Rezumat

Tehnologia informației a produs schimbări semnificative în mare parte dintre industrii. În acest timp, cercetarea recentă privind tehnologia aplicată în educație arată potențialul de a genera un punct de inflexiune care să transforme complet modul în care oamenii învață. Potențiala revoluție educațională actuală este condusă, în principal, de trei componente: inteligența artificială, accesibilitatea internetului și progresele în științele educației. Sistemele inteligente de instruire (denumite prescurtat ITS, din eng. "Intelligent Tutoring Systems"), sunt considerate de experți unul dintre cei mai potriviți candidați pentru transformarea educațională.

Un ITS este un sistem tehnologic care produce instruire personalizată prin material educațional individualizat, solid din punct de vedere pedagogic și ușor de accesat. Un ITS implică studenții în mod independent sau în colaborare pentru a asigura o învățare eficientă. Grupurile de cercetare de Inteligență Artificială în Educație au explorat diverse ipoteze și metode pentru construirea unor sisteme de învățare eficiente în domeniul cognitiv, cu rezultate notabile în discipline precum fizica, matematica și informatica. În schimb, domeniul psihomotric a intrat recent într-un proces intens de digitalizare, și câteva abordări din ultimii ani introduc sisteme de instruire pentru acest domeniu.

Obiectivul general al acestei teze este de a oferi sesiuni de pregătire personalizate în domeniul psihomotric sub forma unui ITS, denumit *Selfit* – un sistem eficient și ușor de utilizat, care are ca obiectiv pe termen lung atât atragerea oamenilor înspre activități sportive, cât și îmbunătățirea sănătății generale a populației.

Acest obiectiv are trei direcții mari de cercetare. În primul rând, introducem o ontologie pentru modelarea conceptelor cheie ale domeniului psihomotric – reprezentările și relațiile dintre concepte, date și entități. Ontologia pentru modelarea cunoștințelor introdusă în cadrul *Selfit*, numită *OntoStrength*, a fost construită folosind metodologia *Ontology Development 101*. Această bază de cunoștințe a fost dezvoltată în cadrul unei echipe multidisciplinare, cu specialiști în știința sportivă, medicină, și ingineria sistemelor software.

În al doilea rând, introducem un algoritm de tip bandit bazat pe context (eng. contextual bandits) pentru generarea de sesiuni personalizate. Procesul decizional al tutorelui digital în fața incertitudinii se dovedește dificil – timpul de pregătire este limitat, studentul nu poate testa toate activitățile sportive, iar personalizarea ar trebui să se întâmple în timp real, menținând în același timp utilizatorul motivat și implicat. Abordarea *Selfit* pentru instruirea automată s-a dovedit, în simulări, mai eficientă ca abordarea de antrenament cu reguli fixe.

În al treilea rând, evaluăm utilitatea și eficacitatea sistemului nostru într-un experiment, ce a implicat 42 de participanți, cu experiență redusă sau medie în antrenamente sportive. Aceștia au primit una dintre cele două versiuni ale sistemului *Selfit*, folosind strategii diferite de adaptabilitate – una restrânsă, iar cealaltă mai largă în dimensiune a explorării. *Selfit* are o interfață mobilă ușor de utilizat, în care utilizatorul poate vizualiza conținutul de antrenament video de executat și trebuie să evalueze efortul implicat de fiecare componentă de antrenament. La sfârșitul experimentului, aceștia au completat un chestionar în care s-a evaluat percepția de utilizare și experiența generală. Utilizatorii au perceput, în majoritate, *Selfit* ca fiind practic, previzibil, simplu, elegant, motivant, nou și captivant. Rezultatele sunt, de asemenea, în concordanță cu simulările noastre inițiale, demonstrând potențialul abordării propuse pentru antrenament personalizat.

Evaluarea sistemului *Selfit*, în ansamblu, a arătat rezultate promițătoare și a evidențiat utilitatea arhitecturii ITS în domeniul psihomotric. Teza actuală poate fi considerată la baza unei noi intersecții disciplinare, ce implică Inteligența Artificială în Educație și pregătirea psihomotorie, deschzând noi direcții de cercetare care vizează îmbunătățirea stării generale de sănătate a populației prin sisteme automatizate.

Cercetări ulterioare pot urmări extinderea bazei de cunoștințe de la antrenamentul de forță la alte tipuri de antrenament, cum ar fi flexibilitatea și mobilitatea, îmbunătățirea experienței utilizatorului prin oferirea de suport vocal în timpul antrenamentului, precum și integrarea tehnicilor de Procesare în Limbaj Natural pentru a îmbunătăți interacțiunea tutore – student, respectiv computer vision pentru evaluare în timp real.

Résumé

Les technologies de l'information ont produit des changements progressifs dans presque toutes les industries. Des recherches récentes sur l'impact de la technologie appliquée à l'éducation ont montré le potentiel de générer un point d'inflexion dans la façon dont les gens apprennent. La révolution éducative actuelle est principalement motivée par trois composantes : l'intelligence artificielle, la disponibilité d'Internet et les progrès des sciences de l'éducation. Les « Intelligent Tutoring Systems » (ITS) sont considérés comme l'un des candidats les mieux adaptés à la transformation numérique de l'éducation. Un ITS est un système informatisé qui produit un enseignement personnalisé grâce à du matériel pédagogique individualisé, pédagogiquement solide et facile d'accès. Les ITS engagent les étudiants de manière indépendante ou collaborative pour assurer un apprentissage efficace. Les groupes de recherche AI in Education (AIED) ont exploré diverses méthodes et hypothèses pour créer des tuteurs efficaces dans le domaine cognitif, avec des résultats notables dans des disciplines comme la physique, les mathématiques et l'informatique. En revanche, le domaine psychomoteur ne présente que récemment un processus de numérisation intensif, car des approches plus récentes introduisent des ITS dans ce domaine.

L'objectif principal de cette thèse est de relever les challenges technologiques associés à la mise en œuvre d'un ITS dédié au développement des compétences psychomotrices dans une perspective de favoriser l'amélioration de la santé générale de la population et de prévenir les accidents du travail. La thèse participe à cet enjeu avec trois contributions. La première est une ontologie dédiée au domaine du développement des compétences psychomotrices. Une équipe multidisciplinaire, avec des spécialistes du sport, de la médecine et de l'informatique a été mobilisée à l'aide de l'approche « Ontology Development 101 » pour formaliser les concepts clés, les représentations et les relations entre les concepts, les données et les entités de ce domaine. La deuxième contribution est un algorithme de la famille des « Contextualized multi-arms bandit » dont la finalité est de générer des sessions de développement personnalisées. Ce processus s'avère difficile. Il existe de nombreuses variables inconnues et des incertitudes : le temps de formation est limité, l'apprenant ne peut pas tester toutes les activités et la personnalisation doit se faire en temps réel tout en maintenant l'utilisateur motivé et engagé. La troisième contribution est le système Selfit, prototype d'ITS dédié aux compétences psychomotrices. Selfit dispose d'une interface mobile conviviale, où l'utilisateur peut visualiser le contenu vidéo de formation à exécuter et doit évaluer l'effort impliqué par chaque composant de formation. L'utilité et l'efficacité de Selfit a été évalué sur une population de 42 utilisateurs, avec une expérience de formation faible et moyenne. Ils ont été impliqués dans une expérience, qui comprenait deux stratégies adaptatives pour le tutorat - une étroite, et l'autre avec une large espace de prospection. Une enquête d'utilisabilité et d'expérience a été remplie à la fin de

l'expérience. Les utilisateurs ont généralement perçu Selfit comme pratique, prévisible, simple, connectif, élégant, motivant, novateur et captivant. Les résultats sont également conformes à nos simulations initiales, démontrant le potentiel de l'approche proposée en formation personnalisée.

Les travaux futurs visent à étendre la base de connaissances de l'entraînement en force à d'autres types d'entraînement, tels que la flexibilité et la mobilité, à enrichir l'expérience utilisateur en fournissant un support vocal pendant l'entraînement, ainsi qu'à intégrer des techniques de PNL pour améliorer l'interaction tuteur-stagiaire et les algorithmes de vision par ordinateur pour permettre une évaluation en temps réel.

List of Abbreviations

AI

AIED

AR

ASR

CBR

FRS

GIFT

GUI

ALT

IRT

ITS

OD101

MDP

ML

NLC

NLP

PTDC

RiARiT

RiERiT

RL

SBTS

SCAMV

SLR

STEM

SWRL

QA

ZPDES

Table of Contents

1	Introduction	23
1.1	Résumé	23
1.2	Goals and Interests	24
1.3	Thesis Outline	26
2	Theoretical Approaches	31
3	Intelligent Tutoring Systems and Psychomotor Training	33
3.1	Résumé	33
3.2	Introduction	34
3.3	Intelligent Tutoring System Architecture and Features	35
3.4	Psychomotor Development in Intelligent Tutoring Systems	39
3.4.1	Main Characteristics of Sports Training	39
3.4.2	Digital Tutors for Psychomotor Development – A Review of the Literature	41
3.5	Challenges in Psychomotor Domain Digitalization	49
3.5.1	Knowledge Modeling	50
3.5.2	Intelligent Tutoring Strategies	51
3.5.3	Interaction with the Trainee	52
4	Models of Knowledge and Learning Process	55
4.1	Résumé	55
4.2	Learning Theories	56
4.3	Modeling Knowledge and Teaching	61
4.3.1	Domain and Student Modeling	62
4.3.2	Teaching Modeling	69
4.3.3	Communication Modeling	78
4.4	Principles of Models Evaluation	87
4.4.1	Tutor Evaluation Principles	89
4.4.2	Evaluation Use-Cases	95
5	Intelligent Tutoring Systems in Open Environments. Machine Learning in Practice	101
5.1	Résumé	101
5.2	Overview of Teaching Techniques for Intelligent Tutors	102
5.2.1	Machine Learning Techniques in Intelligent Tutors	103
5.2.2	Building Reinforcement Learners	107
5.3	Machine Learning Techniques in Psychomotor Development	112
5.4	Examples of Machine Learning-Based Tutors in Other Domains	116
		13

5.4.1	Unsupervised Learning Cognitive Tutors	117
5.4.2	Supervised Learning Cognitive Tutors	118
5.4.3	Reinforcement Learning Cognitive Tutors	122
6	Experiments and Results	129
7	OntoStrength – A Framework to Represent and Inference Knowledge in Psychomotor Intelligent Tutoring System	131
7.1	Résumé	131
7.2	Introduction	132
7.3	OntoStrength’s Design Methodology	132
7.3.1	Design Principles	132
7.3.2	OntoStrength Structure	133
7.3.3	OntoStrength Sub-domains	134
7.3.4	OntoStrength Content	139
7.4	OntoStrength Usage	141
8	Teaching Strategies in a Psychomotor Tutoring System	145
8.1	Résumé	145
8.2	Introduction	146
8.3	Accounting for Sexual Dimorphism in Psychomotor Intelligent Tutoring Systems	147
8.3.1	Basis of Human Sexual Dimorphism	147
8.3.2	Sexual Dimorphism for Scheduling Training Sessions	149
8.4	Optimizing Teaching Sequences using Machine Learning. Right Exercise at the Right Time (RiERiT) Method	150
9	<i>Selfit</i> – A Psychomotor Tutoring System prototype in Open Environments	153
9.1	Résumé	153
9.2	Introduction	154
9.3	<i>Selfit</i> Architecture	155
9.3.1	Domain Module	155
9.3.2	Student Module	159
9.3.3	Tutoring Module	160
9.3.4	Interface Module	161
9.3.5	Sexual Dimorphism Integration. <i>Selfit Version 2</i> Architecture	165
9.4	Workout Generation and Monitoring the Training Impact	167
9.5	Psychomotor Assessment using Computer Vision. A Study on Mitigating the Risk of Injuries	168
10	Results	173
10.1	Résumé	173

10.2	Introduction	174
10.3	Simulation-based Experiment	174
10.3.1	Considering Sexual Dimorphism Dimension in Personalizing Training Sessions	176
10.4	Experiment with Real Users	179
10.5	Results of Agent’s Learning	189
10.5.1	Results for the Acute Effect – Exercise Level Feedback	189
10.5.2	Results on Bandits’ Learning	195
10.6	Results on User Experience	202
10.6.1	Results on User Experience – <i>Selfit version 1</i> – January-February 2021	202
10.6.2	Results on User Experience – <i>Selfit Version 2</i> – January-May 2022	204
11	Discussion and Conclusions	211
11.1	Résumé	211
11.2	Introduction	212
12	Discussion	215
12.1	Advantages of our Approach	215
12.1.1	Knowledge Modeling in Psychomotor Training	215
12.1.2	Teaching Strategies in Psychomotor Training	217
12.1.3	Interaction with the Trainee in Psychomotor Training	220
12.2	Limitations	222
12.2.1	Sample Size	222
12.2.2	Degree of Generalizability	222
12.2.3	Resource	223
12.2.4	Complexity of the Strength Psychomotor Domain	223
12.2.5	Large Codebase	223
12.3	Envisioned Applications	224
13	Conclusion	227
13.1	Résumé	227
13.2	Personal Contributions	228
13.3	Directions for Future Research	231
14	List of Publications	233
15	References	237

List of Figures

Figure 1 Thesis structure	26
Figure 2. A number of scientific articles plotted against the timeline between the years 1980 and 2012, queried using the key phrase "intelligent tutor" in three emblematic bibliographic databases (Ma, Adesope, Nesbit, & Liu, 2014)	35
Figure 3. ITS Four-Component Architecture Overview	36
Figure 4. Super-compensation cycle (Bompa, 2017)	40
Figure 5. Systematic Literature Review - Method Overview	41
Figure 6. Top words in Selected Articles Abstract	49
Figure 7. Disciplines involved in building Intelligent Tutoring Systems	59
Figure 8. Traditional CAI Pedagogical Tree Structure.....	61
Figure 9. GET-BITS model - Main classes used for knowledge representation (Devedzic et al., 2000).....	67
Figure 10. FITS Problem Solving Tutor Architecture (D. Hooshyar et al., 2015).....	71
Figure 11. Authoring Tools used for modeling ITS components (Dermeval et al., 2018)	73
Figure 12. Zone of proximal development evolution according to observed results. (Clement et al., 2015)	76
Figure 13. Apprentice Learner Architecture, as per (MacLellan & Koedinger, 2020).....	78
Figure 14. Virtual Reality Driving Simulator using an Intelligent Tutor (Ropelato et al., 2018).....	81
Figure 15. Santa Tutor Interface – "Estimated Score, Analysis of Problem-Solving Process and Skill Proficiency" (Kim et al., 2020).....	84
Figure 16. ANGELA Main programming concepts mapped to graphical road signs (Schez-Sobrino et al., 2020)	85
Figure 17. ANGELA "Bubble Sort" algorithm visualization using HoloLens in the 3D Environment (Schez-Sobrino et al., 2020).....	85
Figure 18. Evaluation methods used in ITSs between 2007 – 2017 (Mousavinasab et al., 2018).....	88
Figure 19. Evaluation designs schemas	92
Figure 20. PHP ITS. Tutor 1 vs. Tutor 2 Evaluation (Weragama & Reye, 2014)	96
Figure 21. "The frequency of results for the control and the experimental group." (Dolenc & Aberšek, 2015)	97
Figure 22. Skill's Levels Evolution on two simulated populations "P" and "Q" using different teaching approaches (Clement et al., 2015)	98

Figure 23. Virtual Reality Driving Simulator ITS User Experience Evaluation (Ropelato et al., 2018).....	100
Figure 24. Reinforcement Learning Framework. Modeling Agent-Environment Interaction (François-Lavet et al., 2018)	108
Figure 25. Reinforcement Learning Framework. Markov Decision Process. (François-Lavet et al., 2018).	109
Figure 26. The four phases of sports planning.	113
Figure 27. Italk2learn – Bayesian techniques to improve learning ITS goals (Grawemeyer et al., 2016)	118
Figure 28. AITS – Scatter plot of Automated Marker vs Tutor Marker (Grivokostopoulou et al., 2017) ...	119
Figure 29. CRITS – Instructional Cultural Contextualized Architecture (Mohammed & Mohan, 2015)	122
Figure 30. Average accuracy of the formative assessment tutoring (Melesko & Novickij, 2019).....	124
Figure 31. Knowledge matrix and update matrix rules (Andersen et al., 2016).....	125
Figure 32. OntoStrength sub-domain for Strength Skills.....	135
Figure 33. OntoStrength sub-domain for Strength Development Program	137
Figure 34. OntoStrength sub-domain for Personalized Development of Strength Skills	139
Figure 35. OntoStrength MicroCycle Movements query.	140
Figure 36. GraphDB Google Chart SPARQL Query Results for Unique Functional Movements to train in a MicroCycle	140
Figure 37. GraphDB View of Feet Elevated Front Plank instance and class	141
Figure 38. Selfit Graphical User Interface	154
Figure 39. GraphDB View of Micro-Cycle Program Periodization.	160
Figure 40. <i>Selfit</i> Interface –Calibration subcomponent.....	163
Figure 41. <i>Selfit</i> Interface – Training Session subcomponent	164
Figure 42. <i>Selfit</i> Updated Architecture considering Sexual Dimorphism	165
Figure 43. <i>Selfit</i> Interface – Parameters for Workout Generation.....	167
Figure 44. ACL injury risk assessment module process	169
Figure 45. ACL Injury Risk-Assessment Module demonstration	170
Figure 46. Illustration of the evolution of the distance between student’s knee and ankle without valgus.	170
Figure 47. Illustration of the evolution of the distance between student’s knees and ankles with variations between left-and right	171
Figure 48. Training Algorithms Comparison – 2 years timeframe	176
Figure 49. Differences in Training Planification considering Sexual Dimorphism.....	179

Figure 50. Google Analytics Statistics. Users in <i>Selffit</i> between 29th April and 5th May.	182
Figure 51. Training Sessions Count per day and hour slot.	182
Figure 52. Exercises count per training group between 11 th April – 23 rd May.	183
Figure 53. Experiment Statistics on Train Time (a), Muscles to train (b), Material used (c), and RPE (d)..	185
Figure 54. Bandits Exploration Space per each group, based on the Calibration Challenge.....	188
Figure 55. Number of sessions per user and their corresponding training group.....	190
Figure 56. User Interface for Inputting RIR per Exercise type.....	191
Figure 57. RIR values in first mesocycle for (a) Group A and (b) Group B trainees	194
Figure 58. Average Rewards and Moving Average (window size = 6 steps \approx 1 session) Rewards per Bandit Step for Top Performers in Group A (> 11)	196
Figure 59. Average Rewards and Moving Average (window size = 6 steps \approx 1 session) Rewards per Bandit Step for Top Performers in Group B (>2).....	197
Figure 60. Number of Trainees with High Reward (≥ 0.5) per Bandit Step.....	198
Figure 61. Moving Average (window size = 6 steps \approx 1 session) Rewards per User and for Top Performers per Bandit Step in Group A (≥ 12 sessions)	199
Figure 62. Examples of Bandit Choices for User ID #3 based on the available arms.....	200
Figure 63. Bandit Context - User States reported before performing the training session, per each group.	202
Figure 64. Number of training sessions per week reported in the survey	206
Figure 65. Word Cloud with Overall Feedback from User Experience Survey.....	208
Figure 66. Personalized Learning for Psychomotor Training using Contextual Multi-Armed Bandits.....	219

List of Tables

Table 1. Features of Intelligent Tutoring Systems.....	37
Table 2. Scopus search queries.	42
Table 3. ACM Digital Library Search queries with results.	44
Table 4. Springer Search queries with results.	45
Table 5. Database Articles selected after Inclusion / Exclusion Phase	46
Table 6. Articles selected after Quality Assessment Phase.....	47
Table 7. Constructivist-Teaching Methods and Usages, as per Woolf (2010)	59
Table 8. Knowledge Representation in Student Models, as per (Woolf, 2010)	64
Table 9. Features used for aiding the authoring process in ITS Models, as per (Dermeval et al., 2018).....	74
Table 10. Literature review on Natural Language Tutors (1998-2018) – Purpose of the dialogue (Paladines & Ramirez, 2020)	86
Table 11. Evaluation Comparisons Examples and Designs Used with Intelligent Tutors (Woolf, 2010)	93
Table 12. Learners’ evaluation results in the summative assessment. (Dolenc & Aberšek, 2015).....	97
Table 13. The purpose of applying Machine Learning Techniques in ITSs between 2007-2017 (Mousavinasab et al., 2018)	104
Table 14. Number of papers found based on the bandit algorithms used and the and different challenges in adaptive learning (Mui et al., 2021)	110
Table 15. Studies with Smart Sport Tools in the fitness domain (Rajšp & Fister, 2020)	114
Table 16. Upper-Confidence Bound – Simple double assessment model (Melesko & Novickij, 2019)	123
For evaluation, the 400 students have been split into 4 groups: one control group (without the software), one group with Expert Sequence, one group using the ZPDES algorithm, and one group with the RiARiT algorithm. In this study that included pre- and post-tests, all groups participated and the results showed that ZPDES generally performed better than the other algorithms in the real experiments, as indicated in Table 17. Table 17. System Evaluation. Average knowledge acquired per type of algorithm (Clement et al., 2015).....	126
Table 18. OntoStrength RESTful API student’s profiling function.....	142
Table 19. Training load adaptation to menstrual phase adapted from Pitchers	149
Table 20. Micro-cycle Templates Examples for Anatomical Adaptation	150
Table 21. <i>Selfit Domain model</i> . Hip and Knee mapping of body area – movement – muscles involved.....	156
Table 22. <i>Selfit Domain model</i> . Mapping of Movement Areas with Body Areas Movements	158

Table 23. Mesocycle beginner load pattern differences between men and women	177
Table 24. <i>Selfit Version 2</i> mapping of Time Availability and Count of Exercises Proposed.....	184
Table 25. Count of exercises recommended per exercise level.	187
Table 26. Average RIR per exercise and per set with different sessions interval sub-arrays.	192
Table 27. Group A Top Trainers (≥ 12 sessions) Bandit Actions and Sessions Performed.	196
Table 28. <i>Selfit Version 1</i> User Experience Feedback based on AttrakDiff questionnaire.	203
Table 29. <i>Selfit Version 2</i> User Experience Feedback based on AttrakDiff questionnaire.....	205
Table 30. Top ranked features proposed by the users for the next version of <i>Selfit</i>	206

1 Introduction

1.1 Résumé

Ce chapitre est une brève description de la thèse actuelle, avec une lente introduction aux domaines ciblés, qui sont les systèmes de tutorat intelligents, la représentation des connaissances, l'apprentissage automatique utilisé dans l'adaptation des séquences d'enseignement et l'expérience de l'utilisateur. Cette partie commence par une discussion sur l'adoption de la technologie dans la société, qui a été largement responsable de la transformation de notre façon de communiquer et d'apprendre. Un candidat important pour la numérisation de l'éducation est le système de tutorat intelligent, également connu sous le nom d'STI. L'architecture actuelle du système est brièvement présentée dans ce chapitre, y compris la description des quatre composants : Domaine, Étudiant, Tutorat et Modules d'interface.

L'efficacité des STI a été démontrée par le développement de nombreux systèmes ciblant les compétences cognitives, qui ont donné des résultats impressionnants dans l'apprentissage des mathématiques, de la physique ou de l'informatique. Le présent chapitre présente un autre domaine d'apprentissage, le domaine psychomoteur (fondé sur les compétences). Des travaux récents dans ce domaine sont présentés, visant à normaliser et à numériser l'apprentissage psychomoteur. L'objectif de la présente thèse est également présenté : nous visons à construire un STI axé sur l'amélioration de la force et de la santé.

Les principaux défis rencontrés par des recherches similaires dans ce domaine sont décrits en trois volets. D'une part, un défi commun à la construction d'un STI pour un domaine d'apprentissage est la base de connaissances. S'il n'existe pas de travaux antérieurs cartographiant les connaissances spécifiques à modéliser, le coût est élevé et des spécialistes du domaine sont nécessaires pour normaliser et numériser le domaine. Deuxièmement, en fonction de l'objectif pédagogique, un autre défi est représenté par la stratégie d'adaptation du tutorat. Le système automatisé doit être capable d'estimer les niveaux de compétence des étudiants et leur progression pendant l'apprentissage afin d'adapter le matériel et le rythme du tutorat. Troisièmement, l'échange d'informations entre le tuteur et les étudiants doit être bien conçu, facile à utiliser et clair. Le module de communication d'un STI est un élément essentiel et la qualité de l'expérience de l'utilisateur doit être mesurée.

Les trois objectifs de la présente thèse sont mentionnés plus loin, sur la base des défis introduits précédemment. Le plan de la thèse est présenté, décrivant les deux parties principales : Les aspects

théoriques et la partie expérimentale. Ce chapitre se termine par une brève description de chacun des chapitres suivants.

1.2 Goals and Interests

The rise of information technologies, mainly driven by computer innovation, has revolutionized the way we interact and learn (Woolf, 2010). The convergence of Internet science, Synthetic Intelligence, and cognitive sciences has further created new tools in education, improving the way educational content is produced, and delivered, and increasing education efficiency overall. An important candidate in this field is the Intelligent Tutoring System, which is an Artificial Intelligence-based computer system that provides an adaptive educational experience (Fenza & Orciuoli, 2016).

An Intelligent Tutoring System (ITS) aims to enhance student learning experiences by creating immediate, customized instructions and feedback while collecting comprehensive information. Most ITSs are divided from an architectural point of view into four components (Nkambou, 2010), namely:

- *Domain Module* – defines rules, concepts, and problem-solving strategies (expert knowledge);
- *Student Module* – learner’s cognitive and affective states, evolution while learning;
- *Tutoring Module* –selects the best tutoring strategies and actions to take;
- *Interface Module* –responsible for student interaction.

Since its beginning, most approaches for building ITSs targeted cognitive skills – starting with Carbonel in the 1970s (Carbonell, 1970)- and argued for the success of the employed methods in various STEM fields (VanLehn, 2011) and (Butz, Hua, & Maguire, 2006). However, the process of learning involves three domains: cognitive (knowledge-based), affective (attitudinal-based), and psychomotor (skills-based)¹. Recent advancements in software engineering, computer vision, and sports training theory standardization (Bompa, 2017) opened the opportunity for ITS application in the psychomotor skills field.

The present work aims at contributing to the development of ITS in the psychomotor field for large communities of users with a focus on strength and health improvement. Psychomotor skills development is a lifelong process of learning how to move accordingly to a dynamic environment.

¹As established at the Convention of the American Psychological Association in 1918. This classification, based on Bloom’s taxonomy, is considered the de-facto standard (Krathwohl & Anderson, 2001).

A movement competence is a transaction between an individual and a movement task within an environment. Essential movements, such as pushing, pulling, core, knee, or hip-dominant exercises are prerequisites for learning specialized, complex psychomotor tasks required by daily life, professional, or leisure activities. Learning to perform a movement safely and efficiently requires practicing an adequate volume of exercises for enhancing associated physical qualities, such as strength, flexibility, or endurance.

However, one of the main challenges for such a development is the high cost when creating a strong knowledge base from scratch. As stated by Zouaq and Nkambou (2010), an acute research issue on the tutoring module of an ITS is making the design process more efficient and making fit choices regarding knowledge acquisition and representation methods.

In the process of building an ITS, each component is well-defined having roles and rules for implementation. ITS components collectively create a uniform instructional system capable of recognizing student behavior patterns (Orey, 1993) and responding to those patterns with appropriate instructions. However, when it comes to applying it to an open environment (which reflects more realistically the usual training environment), e.g., more unpredictable, and poorly defined, it raises new questions on modeling and efficiency.

Another challenge when developing such a system is to define the right tools and frameworks to acquire accurate student knowledge competencies for predicting its progress while training. Interactive training systems (ITSs) that have been developed for military training (LaViola et al., 2015) are often only used in laboratory settings on standard computers and laptops. These systems typically focus on improving cognitive skills like decision-making and problem-solving, but may not effectively teach physical skills. This could potentially limit the learning and retention of mastering physical tasks.

One of the challenges in building an effective interactive training system is optimizing the teaching sequences to provide the best learning experience for the student. Typically, a tutoring system will use estimates of the student's competence levels and progress to select activities that are most appropriate. However, an ITS that is intended to be accessible to the general public and promote general health should be able to provide a personalized learning experience without requiring a lot of domain knowledge.

A challenge for the tutor is to determine the optimal sequence of activities that will maximize the average competence level for all the targeted skills. (Clement, Roy, Oudeyer, & Lopes, 2015). This challenge, which was raised initially in the cognitive field and has the equivalent in psychomotor development, is driven by three main factors: limited time for practicing activities – the tutor cannot

test all combinations of sequences, or all activities; managing motivation is hard – stimulating activities yield engaged students which learn more effectively; individual differences between students make an optimal sequence for a student inefficient for another one.

In addition, an ITS is deemed to replace the human coach and mediate abstract knowledge with real trainees. Thus, ITS performance is not only determined by the knowledge it carries, but also by the quality of the user experience.

Based on the challenges raised below, this thesis addresses the following major research objectives:

- **RO1:** Design an effective knowledge representation model for the development of the psychomotor skills in an Intelligent Tutoring System.
- **RO2:** Provide personalized sports exercise' recommendations for the mass population when training in open environments.
- **RO3:** Implement an intuitive and effective Communication Module that facilitates the assessment of the sports trainee's progress in open environments.

1.3 Thesis Outline

The structure of the thesis comprises three main chapter groups, *Theoretical Aspects*, *Experiments and Results*, and *Discussions and Conclusions*. Each chapter described in the *Theoretical Aspects* chapter group represents the foundation for the sections presented in the *Experiments and Results* chapter group, as can be seen in Figure 1. The chapters in the *Experiments* part have a corresponding chapter in the *Theoretical Aspects*.

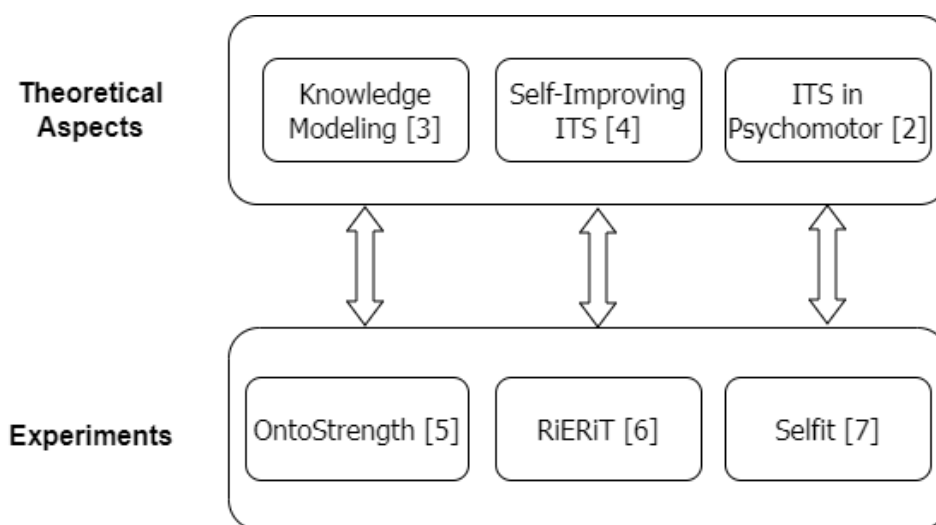


Figure 1 Thesis structure.

The *Theoretical Aspects* structure is the following – first, Intelligent Tutoring Systems are presented, together with their usages in the psychomotor field (Chapter 2), then we present how the knowledge is modeled in ITS (Chapter 3), and what are the methods commonly employed for tutoring in ITSs (Chapter 4).

Then, in the *Experiments and Results* part, we propose a new model for the psychomotor field, called OntoStrength (Chapter 5), which was built based on the findings presented in Chapter 3. Next, a method for personalizing learning sequences in psychomotor training is introduced, called RiERiT – Chapter 6, which was inspired by the findings in Chapter 4, where similar methods were proposed and used in the cognitive field. This chapter also outlines simulations of this method with populations of virtual trainees, and the best converging algorithms are briefly compared and discussed. The *Selfit* system is introduced in the next chapter, as a prototype for psychomotor training ITS in open environments.

Here, the work presented in knowledge modeling and psychomotor tutors' experiments is merged to create a system that showcases the potential of our findings. The *Selfit* chapter also presents the Communication module and how this was integrated to exchange information with the other components. The last chapter in this part introduces the results of our experiments, conducted in two main directions - on one hand, the efficacy of the RiARiT algorithms – bandit learning, and on the other hand, the user's experience. The *Selfit* system was tested with real users, who trained between 1st January 2022 and 31st May 2022, the overall goal being to assess if the proposed prototype is valid for psychomotor training.

Furthermore, the *Discussions and Conclusions* chapter group presents the benefits of our approach, the problems encountered and how they were overcome, and the limitations, along with a list of envisioned applications. The *Conclusions* Chapter presents the summary of the work, our contributions, and potential directions for future research.

Theoretical Approaches

2 Theoretical Approaches

The principles and approaches of technological innovations have driven fast developments or even shifted many sectors in the past 70 years (Chen, Chen, & Lin, 2020). Chen et al. mentioned Henry Ford's analogy on innovation in transportation – sometimes to change an industry it is not enough to do incremental changes, such as making the horses move faster, but to change the whole industry paradigm.

As for transportation, technological advancements have created incremental changes in education too. For a professor, 70 years ago, would have been very hard to see if an essay received from a student may be subject to plagiarism. Also, if for certain reasons, the professor should have left for a few days in another city, he could not perform his job and the course should have been suspended. Now, with modern tools, teachers can video conference into the class and perform their job from any place, if needed.

With the exponential advancements in computers, computing power, and the increasing research and development in the synthetic intelligence field, the field of Artificial Intelligence in Education (AIED) has grown gradually starting with the early '70s. AIED is the field that aims to build computer programs that offer personalized differential teaching, and possesses features that are adaptable to students' responses, and provide optimal teaching sequences (Woolf, 2010). One of the growing fields of AIED is represented by the Intelligent Tutoring Systems, also known as ITS. An ITS is a digitalized educational system deployed on computers, that encompasses intelligence to instruction effectiveness (Grivokostopoulou, Perikos, & Hatzilygeroudis, 2017), to help, improve and, in some cases, replace human teachers' activity. A large majority of ITSs target the cognitive field, in disciplines such as mathematics, physics, or informatics. The first chapter of the current section will discuss the usages of ITSs in the psychomotor field. The psychomotor field has been defined as the second area of learning, by Bloom's taxonomy (Bloom, Engelhart, Furst, Hill, & Krathwohl, 1956), together with cognitive and affective learning.

An intelligent tutoring system should always have information regarding the field of study and the digitally mediated learner and teacher, along with optimal ways of interacting with the learners. The task of describing and modeling such knowledge is challenging. Existing methods and implementations will be described in the second chapter of the current section. Also, an important feature of an ITS is self-improving, which is the ability to learn from experience and offer personalized content to trainees and adapt to new populations, without being explicitly programmed. The self-improving feature will be detailed in the last chapter of the first part.

3 Intelligent Tutoring Systems and Psychomotor Training

3.1 Résumé

Cette section constitue la première partie des approches théoriques. Elle constitue la base du contexte théorique, où les concepts clés sont introduits. Le chapitre commence par un historique du domaine des systèmes de tutorat intelligents, depuis le premier STI construit en 1970 par Carbonell - Scholar, jusqu'à des systèmes de plus en plus évolués. Les systèmes sont devenus de plus en plus complexes et les modules se sont améliorés au fil du temps. Ces améliorations sont allées de pair avec les progrès technologiques, qui ont connu une croissance exponentielle au cours des 50 dernières années.

Une description plus complète est présentée pour chacun des quatre modules, et ce chapitre comprend également une discussion sur ce qui est et ce qui n'est pas un STI. Les principales caractéristiques de tous les systèmes de tutorat comprennent la générativité, la modélisation de l'étudiant, la modélisation de l'expert, l'initiative mixte, l'apprentissage interactif, la modélisation pédagogique et l'auto-amélioration, qui sont brièvement présentées.

Ce chapitre présente également une description du domaine psychomoteur, les principales caractéristiques de l'entraînement sportif et une revue de la littérature sur les tuteurs utilisés pour le développement psychomoteur. La définition des compétences motrices et des règles d'évaluation des stagiaires est présentée. La séance d'entraînement et la méthodologie de régénération sont également présentées et décrites à travers le cycle de super-compensation.

Une revue systématique de la littérature a été menée sur les systèmes de tutorat intelligents pour l'entraînement des capacités psychomotrices en novembre 2019. Le processus d'examen s'est déroulé en trois phases : Identification, Définition des critères d'inclusion/exclusion et Évaluation de la qualité. Les auteurs ont consulté plusieurs bases de données internationales en ligne (Scopus, Web of Science, ScienceDirect, IEEE Explore Digital Library, Springer, ACM et Journal of Education in Data Mining) en utilisant des termes de recherche liés aux STI et aux capacités psychomotrices. L'analyse de la littérature montre que les STI ont également fait leur apparition dans le domaine de la formation psychomotrice, avec des systèmes conçus pour la médecine, l'armée ou les sports.

Enfin, il présente les défis de la numérisation du domaine de la psychomotricité, sur la base des systèmes présentés précédemment. La difficulté de construire des bases de connaissances pour les systèmes de tutorat intelligents est réintroduite et les ontologies sont décrites comme des candidats

pour la représentation des connaissances. Deuxièmement, en ce qui concerne le tutorat dans les environnements ouverts, le défi réside dans le peu d'informations disponibles sur les apprenants et dans la nécessité de trouver un équilibre entre la fourniture d'un apprentissage efficace et la protection de la santé des apprenants et le maintien de leur motivation. Troisièmement, les STI doivent disposer d'un système de communication efficace et simple et ne pas s'appuyer sur les dispositifs IdO pour la collecte de données. Il s'agit d'une tâche difficile, qui implique des efforts considérables de mise en œuvre et de conception.

3.2 Introduction

Intelligent Tutoring Systems (ITSs) are at the cross-road of education and technology (Paviotti, 2012). An ITS is a digitalized instructional system, aimed at supporting learning through different tutoring services that specify what to teach, how to teach, the teaching strategies, and assesses a student's level of mastery on a set of topics for dynamical adaption of content or instructions (Murray, 1999). Development of such systems has been a co-disciplinary process, involving both didactics and knowledge technologies experts. Intelligent tutoring system development calls for an understanding of learning and teaching mechanisms and strategies.

The first intelligent tutor implemented was called Scholar – a system that helps students discover South America's geographical landscapes (Carbonell, 1970), and it was the Ph.D. work of Jaime Carbonell. It was the first work that integrated a semantic network of geography knowledge to propose personalized responses to learners. The Scholar development, along with the developments of John Self, in 1974, 1977, 1985, and William Clancey, in 1979, marked the beginnings of the Artificial Intelligence in Education (also known as AIED) Field (Woolf, 2010).

BIP (Barr, Beard, & Atkinson, 1976) was another instance of an ITS developed in earlier times. The system was able to assign tasks to the learner that were suitable for acquiring competencies in the field of computer programming. BIP programming knowledge representation comprised several specific learned abilities, such as declaring, and printing variables, and the goals were mapped to programming tasks. The student's performance is measured by the accomplishment of these tasks. Each task has a set of skills linked to them which supports inferences on the student profile. The first ITSs built, and many others after had a student model representation structured above or delimited as a subset of the model for the study domain.

In the years after, more intelligent systems were built, either as simulations, or as open-ended learning environments, virtual reality systems, games, or group collaborations, as can be seen in Figure 2. It is important to define what is and what is not an intelligent tutoring system, what it should do and what not, and what are the proposed architectures, rules, common issues, and features. The first part

of the current chapter will focus on describing an overview of the architecture, and the main rules and features for building intelligent tutoring systems.

Of particular interest is to understand what the updates are, if any, on the architecture and rules for ITSs designed to aid the forming of psychomotor abilities, and for this, the second part of the section will present a systematic literature review of this field and the resulting systems. The psychomotor tutoring systems are detailed, with their publication year, the field of interest, architecture, design considerations, and evaluation methods.

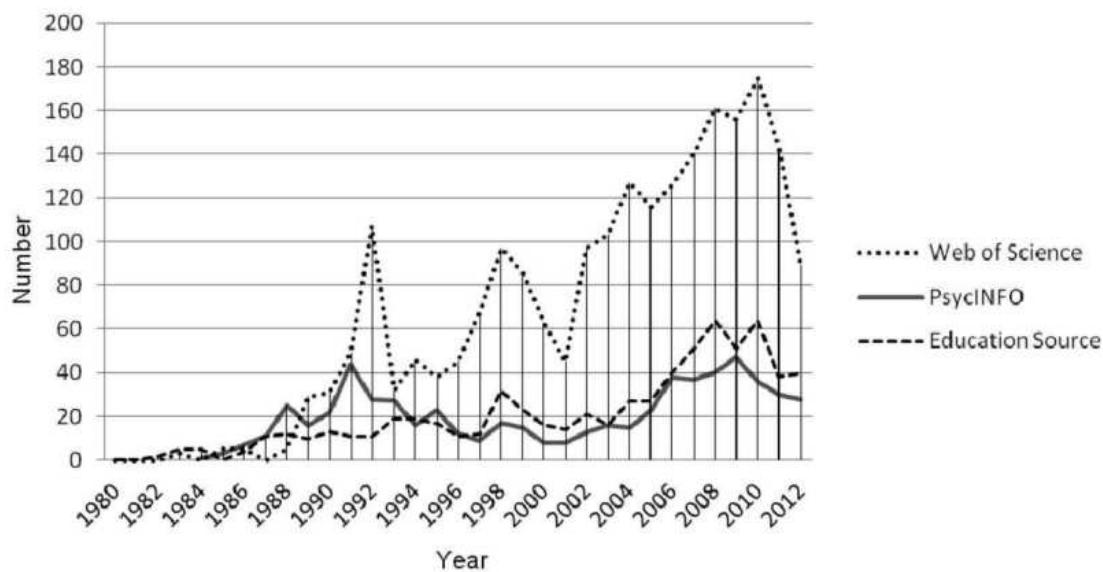


Figure 2. A number of scientific articles plotted against the timeline between the years 1980 and 2012, queried using the key phrase "intelligent tutor" in three emblematic bibliographic databases (Ma, Adesope, Nesbit, & Liu, 2014)

In the last part, existing ITSs in the psychomotor field are critically reviewed to identify problematics in building tutors for the psychomotor domain. The main common challenges are presented, which include knowledge modeling, psychomotor tutoring strategies, and modeling communication with the student.

3.3 Intelligent Tutoring System Architecture and Features

The most common architectural pattern empowered when designing an ITS is the Four-Component Architecture (see Figure 3), which is composed of four modules that comprise the domain representation, tutoring process-related utilities, student-related features, and the GUI (Nkambou, 2010).

The *Domain module* handles knowledge relating to the subject matter and contains concepts, rules, and strategies. ITS uses Domain knowledge to reason with, find solutions to problems or respond

to students' questions. Another feature of the *Domain module* is that it can be used to detect students' errors and propose solutions to correct them. Alternative teaching strategies may be obtained by developing distinct knowledge representations of the same domain knowledge.

The *Tutoring module* provides the necessary knowledge to attain teaching goals. It receives information from both *Domain* and *Student modules*, and it is responsible for selecting the subject content the student will use, providing the response mechanisms for answering the student's questions and patterns to detect when learners need help by embracing different styles of delivery. The *Tutoring module* selects the teaching goals and decides what are the most suitable teaching strategies considering the objectives and the model of the student. A performant *Tutoring module* knows when it should update the process of learning and how to do it. It interacts with the student through feedback and hints.

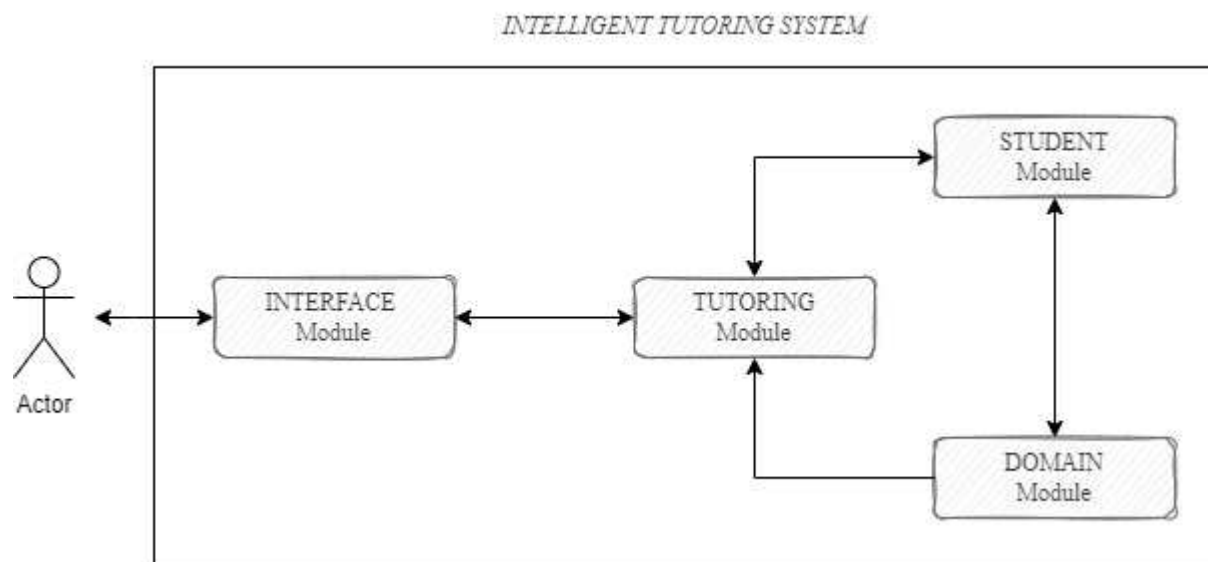


Figure 3. ITS Four-Component Architecture Overview

The *Student module* describes the learner's emerging knowledge and skills, and it is considered a critical component of an ITS. The teaching process should be customized according to the learner's particularities, and based on this, the system needs to collect as much information as it can about learners' preferences, cognitive and affective states, as well as their progress while learning. An ITS is more efficient the more it manages to collect data from and about the learner and use it to perform an analysis of the present knowledge state.

The *Interface module*, also known as the Communication module, or Graphical User Interface module, facilitates the communication between the *Student* and the *Tutoring module*. Even with a well-informed student and knowledgeable teacher, the effectiveness of tutoring can be limited without effective communication strategies. (Woolf, 2010). Therefore, a significant amount of effort should be put

into developing this aspect of the tutoring process. In most cases, the graphical environment is responsible for providing lessons and help while learning, results, and pedagogical actions.

During the steps of building an ITS, each component is well-defined with roles and rules for implementation. ITS components collectively create a uniform instructional system able to identify learner behavior patterns (Orey, 1993) and respond to those patterns with appropriate instructions.

All intelligent tutors are characterized by seven features (Woolf, 2010) – the list can be seen in Table 1, which distinguishes the ITS from the traditional frame-oriented instructional systems.

Table 1. Features of Intelligent Tutoring Systems

Feature	Description
"Generativity"	Represents "the ability to generate appropriate problems, hints, and help customized to student learning needs"
"Student modeling"	Represents "the ability to represent and reason about a student's current knowledge and learning needs and to respond by providing instruction"
"Expert modeling"	Represents the ability to form "a representation and a way to reason about expert performance in the domain and the implied capability to respond by providing instruction"
"Mixed initiative"	Represent "the ability to initiate interactions with a student as well as to interpret and respond usefully to student-initiated interactions"
"Interactive learning"	Represents the ability to generate "learning activities that require authentic student engagement and are appropriately contextualized and domain-relevant"
"Instructional modeling"	Represents "the ability to change teaching mode based on inferences about a student's learning"

"Self-improving" Represents "a system's ability to monitor, evaluate and improve its own teaching performance based on its experience with previous students"

Generativity, also known as "articulate expertise" (Brown, Burton, & deKleer, 1982), refers to the ability of a tutor to generate customized problems, hints, and help on the fly based on the student's knowledge and understanding of the subject matter. In some cases, the tutor may alter or bias the problems in order to increase the likelihood that a specific learning opportunity is presented. This helps the tutor tailor the tutoring experience to the student's needs and goals. If the learner gets stuck, the tutor should provide hints, help, and, optionally, additional problems to move further.

Student knowledge is the dynamically recorded learned tasks based on student action while interacting with the system. Expert knowledge is theoretical framework of the domain which contains relevant information, specific concepts and is organized under different topics. Both student modeling and expert modeling will be described in detail in I.2.1. Domain and Student Modeling section.

The mixed initiative represents the ability to take control of interactions, either by the student or by the tutor. This feature is only partially available in the current ITSs, most of them being driven directly by the tutor. The right implementation of this feature within an ITS would also support students to ask questions to the tutor, but this requires automated understanding and generating responses to students; this task can be achieved using Natural Language Processing techniques (McNamara, Crossley, & Roscoe, 2013).

Interactive learning can be summarized as being constantly responsive to students' learning needs. Achieving an authentic student engagement is a challenging task; it is not enough to just add animations while progressing through the assignments or to visually guide the simulations with graphical elements. Learners need to be continuously motivated to pursue their educational goals. The ITS should first satisfy pedagogical constraints (Gay 1986) and then consider this dimension, as the end goal is to obtain effective learning. The tutor should set as the main priority the learning objectives; this should decide the level of guidance in certain scenarios, and then the interactive learning.

Instructional modeling refers to what type and how much guidance the tutor provides to each student. This should be updated constantly while learning, by receiving feedback from the student. Traditionally, learners with little prior domain knowledge should receive more guidance than the ones with more knowledge. The instruction modeling is responsible for adjusting the guidance level

in the digital tutors, and doing it right has the potential to generate progress in learning outcomes; Shute and Psotka (Shute & Psotka, 1995) have proven that adding instructional modeling and student model in college has shown a boost of 10% in performance.

Self-improving feature refers to the tutor's ability to improve overall system effectiveness according to the past interaction with the learner. This is usually achieved using machine learning or data mining techniques (Clement et al., 2015) that monitor students' progress, judge the efficiency of the prior tutor decisions and adjust future interactions.

3.4 Psychomotor Development in Intelligent Tutoring Systems

According to a meta-analytic review by Kulik and Fletcher (2016), digital tutors have consistently been shown to improve student performance beyond the level achieved in traditional classes and even beyond that of other types of computer tutoring or human tutors. While intelligent tutoring systems (ITSs) have been demonstrated to be effective in supporting cognitive tasks, their potential for supporting the development of other human skills, such as those related to health, is not as well understood. However, ITSs could be highly relevant in addressing health issues.

3.4.1 Main Characteristics of Sports Training

The psychomotor domain, also referred to as psycho-motor or psycho motor or physical, includes physical movement, coordination, and use of the motor-skill areas. The development of skills in this domain requires practice, and the corresponding measurements of performance consider speed, precision, distance, procedures, or techniques in execution.

In the psychomotor domain, several taxonomies were created, among which the most popular ones are Simpson's Taxonomy (Simpson, 1966), useful for the development of children and young people, and Dave's psychomotor domain focused on corporate environments (Dave, 1970). New psychomotor taxonomies were proposed recently, some being more specific (for example, surgical procedures (P. Andreatta, 2019)), and together with the previous traditional ones have been applied for training several psychomotor skills – for example, marksmanship training for the US Army (D Brown, 2017).

A movement competence is a transaction between an individual and a movement task within an environment. Essential movements such as pushing, pulling, core, knee, or hip-dominant exercises are prerequisites for learning specialized, complex psychomotor tasks required by daily life, professional, or leisure activities. Learning to perform a movement safely and efficiently required

practicing an adequate volume of exercises for enhancing associated physical qualities such as strength, flexibility, or endurance.

Psychomotor development usually starts with the definition of movement competence and the initial evaluation of students. The tutor then defines learning objectives, orientation, methods, and means and generates a first training program. At each milestone of the program, the tutor compares real and predicted results and updates the program if necessary. Students should have enhanced their psychomotor capacities to perform targeted movements safely and efficiently at the end of the program.

The development of a physical quality requires following the super-compensation cycle (Bompa, 2017). This cycle associates effort and strengthening that make up the process of physical adaptation and progression. Exercises intensities, volume, and bio-energetic specificities must be systematically and rationally alternated in a sequence of four phases (cf. Figure 4): a one to two hours phase of central and peripheral fatigue generated by practicing physical exercises; twenty-four hours to forty-eight hours phase of restoration and compensation; a thirty-six to seventy-two hours phase of physiological supercompensation and a three to seven days involution phase where physiological benefits of supercompensation decreased.

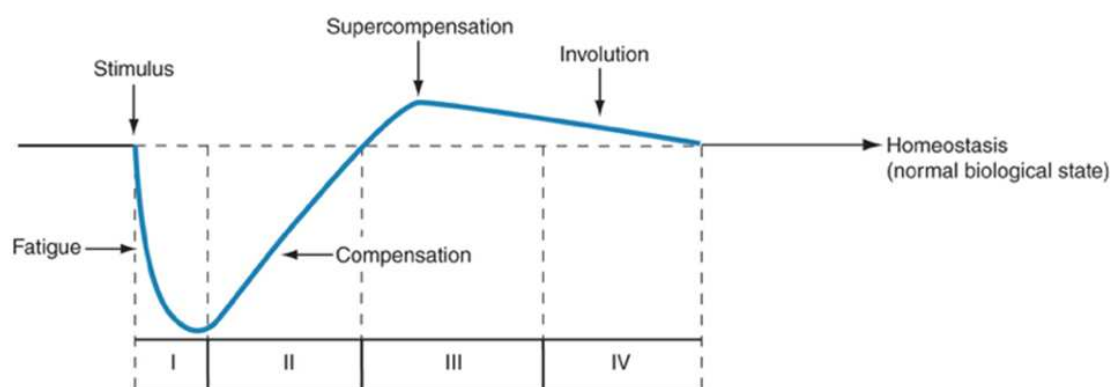


Figure 4. Super-compensation cycle (Bompa, 2017)

Learning how each student reacts to stimuli will support the initialization and the update of parameters structuring the different temporality of a physical qualities' development program: macrocycle (8 – 12 weeks), mesocycle (2-4 weeks), micro-cycle (7 days), session (30-120mn), set of exercises (duration and intensity, grouped by the number of repetitions) to maximize super-compensation and prevent negative interferences between stimuli. Intelligent Tutoring Systems (ITS) and Machine Learning can bring added value to optimizing psychomotor development.

3.4.2 Digital Tutors for Psychomotor Development – A Review of the Literature

A systematic literature review (Neagu, Rigaud, Travadel, Dascalu, & Rughinis, 2020) was conducted in November 2019 on the most reputable online data sources for determining quantity and value of the studies that target the conceptualization and implementation of ITSs for improving psychomotor capabilities. This work was the foundation for the experiments presented further in this thesis.

The method to identify the existing digital tutors for psychomotor development implied three phases – Identification Phase, Inclusion/Exclusion Criteria Definition Phase, and Quality Assessment Phase, as can be seen in Figure 5.

The included papers after the review rely on ITS architectures and are used in training psychomotor skills. All of the accepted systems were tested and have proven results in their specific domains.

A Identification

A.1 Search Query

The selected search terms aim to map the articles which refer to both Intelligent Tutoring Systems and psychomotor abilities. Searches also included synonyms for ITS to capture all the other terminologies: Knowledge-Based Tutoring System, Intelligent Computer-Aided/-Assisted Instruction, Computer-based Tutoring System, and Adaptive Tutoring System. Psychomotor abilities are referred to sometimes as "psycho-motor" or "psycho motor", and even "physical", a broader term that maps the activities of a person.

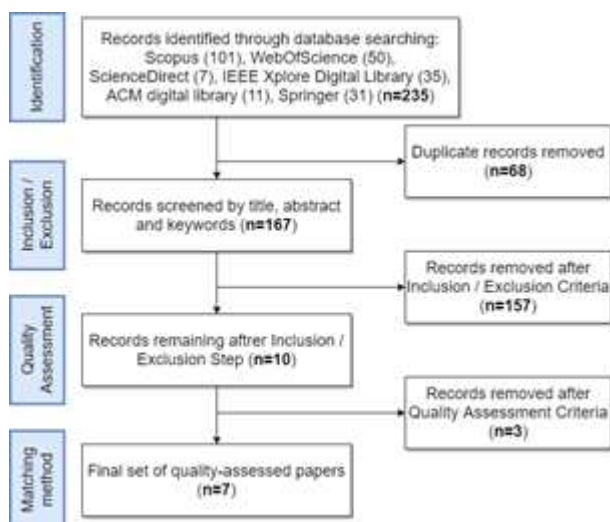


Figure 5. Systematic Literature Review - Method Overview

The search query applied for obtaining the list of articles in the online databases is the following:

("Intelligent Tutoring System*" OR "Intelligent Computer-Aided Instruction*" OR "Intelligent Computer-Assisted Instruction*" OR "Knowledge-Based Tutoring System*" OR "Adaptive Tutoring System*" OR "Computer-based Tutoring System*") AND (psycho-motor OR psychomotor OR "psycho motor" OR physical)

Even though the above query was the targeted search query to be applied to all the data sources, different constraints were encountered while using each search engine, such as the length of the query provided was too large, or too many masks were applied (where the mask is considered "*" character), all presented further in detail.

A.2 Data Sources

The literature review was conducted in November 2019 using the following electronic international databases: Scopus, Web of Science, ScienceDirect, IEEE Explore Digital Library, Springer, ACM, and Journal of Education in Data Mining. These data sources are the most common online sets used in scientific research (Dieste, Grimán, & Juristo, 2009). Detailed results returned while searching the relevant articles in targeted databases are presented below, showing queries adapted for each of them, the total number of articles and duplicates found.

Scopus². The query was applied in the title, abstract, and keywords search sections. The mapping of all the keywords in only one clause generated an error: the size of the search query was too large. The initial query was split into smaller sub-queries and executed one by one. Sub-queries with the resulting number of papers greater than zero and duplicates are presented in Table 2.

Table 2. Scopus search queries.

Search query	# Articles	Duplicates
«TITLE-ABS-KEY("Intelligent Tutoring System*" AND (psycho-motor OR psychomotor OR "psycho motor" OR physical)) »	98	43
«TITLE-ABS-KEY("Intelligent Computer-Aided Instruction*" AND (psycho-motor OR psychomotor OR "psycho motor" OR physical)) »	1	1

² <https://www.scopus.com/search/form.uri?display=basic&zone=header&origin=>

«TITLE-ABS-KEY("Intelligent Computer-Assisted Instruction*" AND 2 0
(psycho-motor OR psychomotor OR "psycho motor" OR physical)) »

«TITLE-ABS-KEY("Knowledge-Based Tutoring System*" AND (psycho- 0 0
motor OR psychomotor OR "psycho motor" OR physical)) »

«TITLE-ABS-KEY("Adaptive Tutoring System*" AND (psycho-motor 0 0
OR psychomotor OR "psycho motor" OR physical)) »

«TITLE-ABS-KEY("Computer-based Tutoring System*" AND (psycho- 0 0
motor OR psychomotor OR "psycho motor" OR physical)) »

The term "Intelligent Tutoring System" is by far the most common, while "physical" was the most used term in the right part of the query. The total number of articles returned from the Scopus search was 101, of which 44 papers were duplicates.

WebOfScience³. The advanced search option enables querying the database by different fields, such as topic, title, author, editor, publication name, conference, or publishing year. As there is no direct mapping with title, abstract, and keywords specifically, the closest functionality was to search by topic (referred to as "TS"). The query used for searching articles in this database was the following:

«TS = (("Intelligent Tutoring System*" OR "Intelligent Computer-Aided Instruction*" OR "Intelligent Computer-Assisted Instruction*" OR "Knowledge-Based Tutoring System*" OR "Adaptive Tutoring System*" OR "Computer-based Tutoring System*") AND (psycho-motor OR psychomotor OR "psycho motor" OR physical)) »

Starting with the query above, the number of articles returned was 50 using all the available indexes (SCI-Expanded, SSCI, A&HCI, CPCI-S, CCR-Expanded, etc.), throughout all indexed years (from 1975 to 2019) and in all available languages. Of the returned 50 articles, 34 were duplicates from other databases.

ScienceDirect⁴ includes advanced search functionality that finds a query within "Article title, Abstract, Keywords", the latter being the ones specified by the author. The initial search query was too large to be applied in the database, so it was split down into subqueries based on the terms which are referring to Intelligent Tutoring Systems, similar to the Scopus search. Terms referring to ITS

³ <https://webofknowledge.com>

⁴ <https://www.sciencedirect.com>

were paired with sub-query « (psycho-motor OR psychomotor OR "psycho motor" OR physical) » and the search returned no results. Only the term "Intelligent Tutoring System", paired with the above sub-query returned seven entries, from which six were duplicates. In ScienceDirect, the wildcard "*" is not supported and we used both singular and plural versions of the terms; however, the same results were returned.

IEEE Xplore Digital Library⁵ has an advanced search option that lets users search in Metadata, or Full text and Metadata. Searching rules are mainly driven by the website interface in IEEE, and the searching strategy was the same as for Scopus: applying the terminologies used for Intelligent Tutoring Systems to all the definitions used for psychomotor activities. Nevertheless, the search URL was modified to match the current request due to the restrictions in the platform interface to search directly for a specific query. The sub-queries were applied for each of the terms used to refer to ITS, but only "Intelligent Tutoring System*" returned at least one entry.

«("Intelligent Tutoring System*" AND (psycho-motor OR psychomotor OR "psycho motor" OR physical)) »

The sub-query above returned 35 papers, of which 17 were duplicates. All the other sub-queries, matching other terminologies (i.e., "Intelligent Computer-Aided Instruction*", "Intelligent Computer-Assisted Instruction*", "Knowledge-Based Tutoring System*", "Adaptive Tutoring System*" and "Computer-based Tutoring System*") returned no result.

ACM Digital Library⁶ is similar to IEEE Xplore. Users can search individually in the title or the abstract, but there is no way to search in the keywords. As more complex queries were hard to compute from the platform, we grouped search words two by two (e.g., "Intelligent Tutoring System AND physical", or "Intelligent Computer-Aided Instruction AND psychomotor"). The sub-queries with results are presented in Table 3.

Table 3. ACM Digital Library Search queries with results.

Search query	# Articles	Duplicates
«recordAbstract:(+physical +"Intelligent Tutoring System") »	10	2

⁵ <https://ieeexplore.ieee.org/Xplore/home.js>

⁶ <https://dl.acm.org>

«recordAbstract:(+physical + "Intelligent Computer-Aided Instruction") » 1 1

Springer⁷. The advanced search functionality on the platform was not returning the correct results at the time of the search. Instead, the search was performed using a scrapping tool developed by Google, called inURL⁸. As for the ACM platform, search words were grouped two by two. The inURL results which returned at least one article are presented in Table 4.

Table 4. Springer Search queries with results.

Search query	# Articles	Duplicates
"Intelligent Tutoring System*" "physical" inurl: springerlink.com	27	8
"Intelligent Tutoring System*" "psychomotor" inurl: springerlink.com	1	0
"Intelligent Computer-Aided Instruction*" "physical" inurl: springerlink.com	1	0
"Intelligent Computer-Assisted Instruction*" "physical" inurl: springerlink.com	2	0

The above queries were applied in the Google Search engine, and the inURL parameter was applied to search within the entire content of pages on SpringerLink, not only the title, abstract, and keywords. Taking into consideration that the number of results was not too large (31 entries), all articles were taken into consideration at this step.

B Inclusion / Exclusion Criteria

B.1 Inclusion criteria

The literature review work was focused on finding adequate papers with original research on different areas where ITSs were used for training psychomotor abilities. Thus, the following inclusion criteria were developed:

⁷ <https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines>

⁸ https://en.ryte.com/wiki/InURL_Search

1. Full papers and peer-reviewed papers;
2. Papers with empirical research (qualitative and quantitative);
3. Papers describing ITS architectures or variations;
4. Papers clearly explain areas of the psychomotor domain, where ITS was applied.

Criteria 1 and 2 were chosen to ensure that our method focuses only on high-quality work and original research. Criterion 3 was included in determining ITS architectures used as main starting points in papers, and not only references to such systems. Lastly, we focus on papers where applications of the psychomotor domain are presented.

B.2 Exclusion criteria

The method reiterates the focus on quality work, which references usages of ITS in psychomotor areas. Criteria were chosen to exclude duplicate papers or papers which were initially work-in-progress and then, later on, fully reported and published. The exclusion criteria include the following:

1. Study protocols and extended abstracts;
2. Papers referencing ITS, but do not use specific architectures, or variations;
3. Papers referencing the psychomotor domain but applying ITS to different domains.

The first criterion excludes the incomplete versions of papers, papers that are not peer-reviewed, or study cases. Criteria 2 and 3 ensure that ITS architectures or small variations are used for applications in the psychomotor domain.

B.3 Results

Table 5 centralizes the results after the inclusion/exclusion steps and depicts the papers considered relevant for in-depth follow-up analyses. Articles considered valid at this step describe ITS architectures (or several variations) and are applied to different areas of the psychomotor domain, from training military skills to medical training or driving a car.

Table 5. Database Articles selected after Inclusion / Exclusion Phase

Database	# Articles	Duplicates	Valid after the I/E step
Scopus	101	44	10
Web of Science	50	34	0

ScienceDirect	7	6	2
IEEE Xplore Digital Library	35	17	3
ACM Digital Library	11	3	0
Springer	31	8	1

The papers related to psychomotor training vary from very specific (e.g., training for marksmanship, training for ball-passing), to broader activities (e.g., improving human motor learning, training physical tasks). The papers matching the Inclusion / Exclusion criteria were published at several conferences in a timeframe of twelve years, while most papers were published in the past two years.

C *Quality Assessment*

The next step implied reading the accepted papers after inclusion/exclusion filters and assessing if the solutions present any outcomes of the proposed architectures and if they are following any psychomotor taxonomies. The final list of papers matching our criteria contains seven articles, presented with details relevant to the current research in Table 6.

Table 6. Articles selected after Quality Assessment Phase

Paper name	Year published	Architecture	Psychomotor Domain	Results
"Development and application of a multi-modal task analysis to support intelligent tutoring of complex skills" (Skinner et al., 2018)	April 2018	Intelligent tutoring system	Training Robotic-Assisted Laparoscopic Surgery	A multi-modal task analysis (MMTA) is applied and it has shown effectiveness for training complex skills.
"Modeling expert behavior in support of an adaptive psychomotor training environment: a marksmanship use case"	September 2017	Generalized Intelligent Framework	Training Rifle Marksmanship	Presented procedures of measuring, creating models, and validation from experts in the psychomotor domain,

(Goldberg, Amburn, Ragusa, & Chen, 2018)		k for Tutoring		which lead to skill acquisition and perfection
"TUMA: Towards an intelligent tutoring system for manual-procedural activities" (Hodaie, Haladjian, & Bruegge, 2018)	May 2018	Intelligent tutoring system	Manual-procedural activities	Vision, requirements, functional architecture, research road map
"Adaptive tutoring on a virtual reality driving simulator" (Ropelato, Zund, Magnenat, Menozzi, & Sumner, 2018)	January 2018	Intelligent tutoring system	Driving	Five categories of activities related to vehicle driving are aided by the system in order to train and evaluate the learner automatically
"Intelligent augmented reality tutoring for physical tasks with medical professionals" (Almiyad, Oakden-Rayner, Weerasinghe, & Billinghamurst, 2017)	June 2017	Intelligent tutoring system	Percutaneous radiology procedures	Assist junior radiologists in achieving competency in image-guided procedures
"A framework for an interactive robot-based tutoring system and its application to ball-passing training" (Lee & Kim, 2010)	December 2010	Interactive Robot-based Tutoring System	Ball-passing training	Robots can deliver tasks that are adjusted for the user's individual needs and enhance their ball-passing skills by offering feedback
"TIKL: Development of a wearable vibrotactile feedback suit for improved human motor learning"	October 2007	Tactile Interaction for Kinesthetic Learning	Sports training, dance, postural	The learner exhibits considerably enhanced accuracy when executing the indicated movement,

(Lieberman & Breazeal, 2007)

retraining in with a remarkably faster health rate of skill acquisition

Even though there are not many ITS used for training psychomotor skills, their areas of usage are broad. Words like "intelligent", "tutoring", "system", "training", or "models" are some of the most common terms, with more than one appearance on average per paper. Topics like "feedback", "cognitive" and "physical" are also used often, while more specific terms for each sub-field in psychomotor training, such as "marksmanship", "radiology", "driver", "procedure", "angle", are not frequent in all retrieved article. The full distribution of words based on frequency is presented in Figure 6.



Figure 6. Top words in Selected Articles Abstract

From this literature review, we conclude that, even though initially developed for training cognitive skills, ITSs have emerged also in psychomotor training, with usages in several sub-fields, such as Medicine (Laparoscopic Surgery Training, Radiology), Military (GIFT), Driving (Adaptive VR driving simulator), Sports (Football - Ball-Passing Training, dance, retraining for health), and Generic manual procedures (TUMA).

3.5 Challenges in Psychomotor Domain Digitalization

Literature in the field of ITS applied to psychomotor training is not vast, even though last years show an increasing interest in the field, 5 of the 7 systems identified were built in the last 5 years. Even though they target psychomotor development, an in-depth analysis showed that, for most of them, there was no clear or detailed explanation of how the ITS modules are impacted and modified by their usage in a non-cognitive domain.

In addition, when designing ITSs in open environments there is a high level of uncertainty to face when assessing progress and while tutoring. Specific rules for unveiling the learner profile while training are required with a keenness on the learner's medical shape to prevent injuries or health issues during workouts.

The trainee is usually in-the-wild, with limited time, and the system is responsible for maintaining the trainee's motivation to achieve his training goals. The communication layer between the system and the trainee needs to be clear, simple, and efficient. The above challenges in designing a Psychomotor ITS in open environments will be detailed further.

3.5.1 Knowledge Modeling

One of the most difficult challenges when building ITSs is creating a strong knowledge base from scratch (Zouaq & Nkambou, 2010). The tasks for representing the knowledge, the knowledge acquisition techniques, and efficient modeling of the tutoring require acute research in the field.

Sports training periodization is considered a young field. The first research work on describing the rules for training has been performed 5 years ago by Bompa (2017). Before there were no clear standards for performing sports training and periodization, and different sports coaches were using different terminologies when referring to specific topics.

The knowledge representation has been implemented traditionally in an ITS through either black-box models, or glass-box models. Both models include the following types of knowledge: declarative (conceptual information), or procedural (for action sequences and problem-solving procedures) (Zouaq & Nkambou, 2010). The advancement of the Semantic Web also in the Educational field (Aroyo & Dicheva, 2004) called for uniformity between the approaches and the opportunity to reuse existing knowledge in different systems. This enabled the development of ontologies, which is a shared vocabulary and representation of knowledge, well fitted to support data integration, modularity, and standardization (Fensel, 2001).

Based on these considerations, ontologies are becoming a strong candidate for building knowledge in Intelligent Tutoring Systems and adaptations of such systems. As stated by Neagu, Guarnieri, et al. (2020), no previous ontology has been developed in the psychomotor field; existing works focus on recognizing sports activities with technology or support decision-making based on data collected during sports competitions. The effort of building psychomotor development knowledge requires multidisciplinary expertise, and good coordination for defining concepts and relations in learning branches such as physiology, biomechanical, muscular, or medical areas.

Except for the GIFT project (Brawner, Hoffman, & Nye, 2019), the other existing work in psychomotor ITSs does not clearly outline how the knowledge is modeled. Cross-disciplinary teams of the GIFT NATO department developed and described the knowledge modeling for training

marksmanship at the GIFT Symposium⁹ every year, proposing also recently ontology-driven approaches for the modeling (Brawner et al., 2019).

3.5.2 Intelligent Tutoring Strategies

The tutoring module of an ITS is responsible for selecting the optimal learning sequences in order to provide a good learning experience for the student (Clement et al., 2015). This is based on estimates of the student's competence levels and progress, as well as limited knowledge about the cognitive and student models. The same principles are applied in the psychomotor field too. The literature shows that for learning how to drive, Ropelato et al. (2018) created a virtual reality environment where learners receive optimal sequences based on the Zone of Proximal Development and Empirical Success (ZPDES) algorithm. A limited set of actions are defined to be learned while following driving simulations: stable driving (on straight roads and curved roads), turning, complete stop, constant speed (without elevation and on up or down hills), and reaction in traffic. The activities were represented in an activity graph, based on their difficulty, and the ZPDES algorithm was used to generate the next optimal activity.

The ball-passing training ITS (Lee & Kim, 2010) has no tutoring adaptation. Learner performance was measured with sensors attached to a robot tutor while training. The tutoring process for this training involves both rule-based and score-based approaches to ball passing. The rule-based approach is suitable for novice learners, as it relies on expert opinions about the user's performance. Score-based training is more appropriate for semi-expert learners, as it is based on the user's own performance. Both of these approaches can be useful in helping the learner improve their skills.

TUMA system (Hodaie et al., 2018) is in the projection phase, and the authors challenge the adaptive component for manual-procedural activities. They outline the importance of self-improving tutors, but without describing a clear perspective for their future implementation.

The rifle marksmanship tutoring process focuses on teaching the fundamental skills needed to effectively use the weapon. The instructions are primarily focused on helping the learner consistently hit static targets at fixed distances. This training is important for ensuring that the learner is able to accurately and effectively use the rifle (Goldberg et al., 2018). In the closed environment, the training components: stability, aiming, control, and movement, are tracked by the tutor using sensing technologies. The model for personalizing the learner training sequence tracked parameters such as the position of the body, the inspiration/expiration, the pull on the trigger, and attaching the weapon

⁹ <https://gifttutoring.org/projects/gift/wiki/Overview>

silencer while training, but the algorithms implemented for adjusting the sequences were not clearly described. The other ITSs used for training psychomotor skills which were found in the literature review presented above did not show any adaptive behavior of the tutor. The systems presented above are used in closed environments, where tutors have control and usually comprehensive knowledge of trainees' status and learning progress.

Self-improving tutors are becoming increasingly popular in ITSs for a number of reasons. These include increased tutor flexibility, reduced long-term tutor cost, ease of adaptation to new student populations without human intervention, increased generality, and the ability to reason about uncertainty (Woolf, 2010). These characteristics make self-improving tutors an attractive option for use in ITSs.

A big challenge in tutoring in open environments is caused by the lack of accurate learner information while training. Also, time resources are limited for the learners, a learner cannot test all the training activities or all the existing sequences of training. A rule-based tutor may prove inefficient both in the cognitive and psychomotor fields. The order and length of the learning steps for one trainee may not be efficient for another one (Clement et al., 2015). Building a psychomotor tutor in an open environment is a challenging task, as minimum learner data should be acquired while training, tutoring relies on the trainee's input, partially acquired data from the environment, and caution on learner's health— new sequences generated by the adaptive tutor should protect the learner from injuries, medical issues, or physical exhaustion. On the other side, the learner should be engaged in learning and maintain long-term motivation for training. Learning should be delivered in the trainee's zone of proximal development.

3.5.3 Interaction with the Trainee

A tutor may have the best student and teaching knowledge, but without an efficient interface component to interact with the learner, it will limit the value of the tutor (Woolf, 2010). The user interfaces should be clear and simple, attractive, and responsive. Based on these considerations, building a proper communication module usually involves a big amount of effort. A human tutor easily detects learner reactions in the classroom, he can easily detect problems and provides feedback and remediation. The traditional tutor can track learners' focus of attention during classes, level of fatigue, and motivation. The physical trainer acts similarly. Trainees are monitored closely, before, during, and after workouts, with or without the help of sensing technologies.

The classification of the communication layers between tutors and learners inside an ITS includes graphics communication, social intelligence, component interfaces, and natural language techniques (Woolf, 2010). Graphics communication refers to either emulating synthetic humans to engage in

conversation with learners, creating virtual environments for learning, or other sophisticated graphics techniques (facial animations, special effects, or artificial life). Social intelligence refers to components that may real-time assess a learner's state, through visual recognition of emotion (real-time camera on the learner to detect facial emotion), or metabolic indicators using sensors for posture, mice pressure, and speech recognition technologies. Component interfaces are responsible for processing student input and providing feedback. Natural Language Communication implies advanced natural language processing techniques through which the tutor can understand and generate written and spoken dialogue.

In the psychomotor field, the Driving Simulator ITS uses a virtual reality environment (Ropelato et al., 2018) to interact with the user and artificial intelligence to automatically generate the city. The city includes 3D models for cars, roads, and traffic simulation. The learning environment is closed, and the learner has a special setup where he trains – a room for training, with a motion system, a driving seat, a steering wheel, pedals, 3 monitors to display the virtual environment, and a strong computer to render the 3D models at 60 Frames per second.

Ball-pass training interaction was facilitated by a robot tutor, which is a physical robot equipped with an omnidirectional vision system and mobility. (Lee & Kim, 2010). The robot was able to measure the angle, speed, and distance of an approaching ball. During training, the robot would take a position on the pitch and wait for the learner to initiate a pass. The specific tasks and training activities were determined based on the needs of the learner. After getting the ball, the robot passes it back to the learner and provides feedback on the hit. One disadvantage of this approach is the high cost of developing and deploying robot tutors for many learners.

The TUMA (Hodaie et al., 2018) system was designed to provide support and guidance for manual procedures. This included a communication module that provided step-by-step instructions, as well as tracking of hand movements, gestures, and tool usage to evaluate their correctness. The goal of TUMA is to help learners understand and perform manual procedures accurately and effectively. The psychomotor ITS proposes a closed environment, in a well-configured room, where a camera supervises the movements, and the instructions are projected on a table.

Marksmanship training (Goldberg et al., 2018) is conducted in a simulated environment using a simulated M4 carbine equipped with sensor technologies to monitor the learner's behavior. An additional breathing sensor was also used to monitor electrocardiogram signals and breathing waveforms. The GIFT system, which is a variation of an ITS, includes a data capture and logging layer that stores key data during training, such as information about new targets, aim data, weapon

sensor data, and fire events. This helps the trainer understand the learner's performance and identify areas for improvement.

In an open environment, the interaction with the learner proves to be more challenging than in closed, fully supervised environments. The learner training setup is unknown, and assessment is hard to perform. For learning psychomotor skills, the learner needs guidance for execution, which can be video, image, or written, and voice or visual feedback. Also, the assessment may prove critical for both tracking the learner's progress and avoiding the risk of injuries. The more data the system gathers, the better the assessment and personalization should be.

The data used to measure the training impact, such as heart rate, calories burnt, or quality of sleep, which can accurately be gathered through IoT devices (smartwatches, smart bands) should not be mandatory and just enhance the model, but it should not rely on them. The communication between the system and the trainee needs to be efficient and simple, the trainer should not be forced to be tied up with the system while training. Designing such a system is a challenge that, from our findings, was not addressed in the literature.

The literature review shown an increasing interest of the research communities in designing and developing intelligent tutors for psychomotor training. Researchers introduced in the past years systems in several psychomotor sub-fields, including ball-passing training, medical (for surgeries), car driving, or military (marksmanship training). Our goal is to build a psychomotor tutor for athletization, and to provide optimal training sessions for novice and intermediate trainees.

We defined a main set of challenges encountered while building such a system, which is also mapping the ITS architecture modules – the domain, student, tutoring, and interface. The challenges were defined in three directions: (1) Knowledge modeling, and a more comprehensive description will be presented in Chapter 4, (2) Intelligent Tutoring Strategies, which will be elaborated in Chapter 5, and (3) Interaction with the Trainee, where methods for efficient communication between tutor and the learner are introduced in Chapter 4.3.3 and our proposal for psychomotor training will be presented in Chapter 9.

4 Models of Knowledge and Learning Process

4.1 Résumé

Ce chapitre examine tout d'abord la différence entre les philosophies d'enseignement traditionnelles et constructivistes. Les systèmes tutoriels intelligents (STI) sont utilisés pour soutenir l'apprentissage, mais la plupart d'entre eux utilisent la philosophie d'enseignement traditionnelle. L'enseignement constructiviste est plus difficile à mettre en œuvre et est limité par les ressources et le temps nécessaires, ainsi que par son adéquation aux environnements technologiques. Certains STI ont été conçus selon l'approche constructiviste, mais une validation plus poussée est nécessaire.

Ensuite, une description des techniques de modélisation des connaissances et de l'enseignement est présentée. Le passage de l'enseignement assisté par ordinateur à l'STI met en évidence les avancées entre les deux types de tuteurs. Les STI peuvent diagnostiquer avec précision le niveau de connaissances des étudiants et décider de ce qu'il faut faire ensuite en fonction du niveau de connaissances estimé. En outre, un STI peut fournir un retour d'information aux apprenants.

Ce chapitre présente également des techniques courantes de génération de modèles d'apprenants : Réseaux bayésiens, réseaux de croyances, raisonnement à partir de cas (CBR) et maximisation des attentes. Les formalismes de représentation sont brièvement présentés et les modèles de représentation des étudiants sont décrits : superposition sur le modèle de l'étudiant, bibliothèque de bogues, bande passante et ontologies. Une ontologie de domaine est une alternative émergente pour la représentation des connaissances lors de la construction de STI. Cela a été prouvé par certains STI récents qui ont été construits plus rapidement, sont interconnectables, interrogeables et normalisés par le biais d'un vocabulaire partagé. Les méthodologies de représentation des connaissances sont présentées, parmi lesquelles un candidat bien établi est *Ontology Development 101 (OD101)*. Le langage de représentation formelle *OWL* est également présenté, ainsi que l'outil utilisé pour la construction d'ontologies, *Protégé*.

Les STI visent à simuler les tuteurs humains dans l'enseignement et à fournir des expériences d'apprentissage personnalisées. Les stratégies de tutorat dans les STI peuvent être basées sur les observations d'enseignants humains, sur des théories d'apprentissage ou sur des approches technologiques. Par exemple, les tuteurs de résolution de problèmes sont couramment utilisés dans l'enseignement des mathématiques, de la physique ou de la programmation informatique. Le choix de la théorie d'apprentissage utilisée dans les STI dépend du domaine enseigné, de la nature de l'apprentissage et du niveau des apprenants. Les théories d'apprentissage cognitif couramment

utilisées dans les STI comprennent le constructivisme, l'apprentissage systématique et l'apprentissage social. La stratégie d'enseignement de l'apprentissage repose sur l'idée que les apprenants acquièrent des compétences par le biais d'une interaction sociale avec des experts. L'objectif est de développer des modèles mentaux solides grâce à une reproduction réaliste du processus d'apprentissage. Plusieurs systèmes sont présentés plus loin, avec le domaine d'apprentissage ciblé, les détails de la mise en œuvre et l'évaluation.

La composante communication d'un tuteur intelligent joue un rôle crucial dans son efficacité. Des techniques telles que les tuteurs animés, la réalité virtuelle, la reconnaissance visuelle des émotions et le traitement du langage naturel sont utilisées pour modéliser la communication. Les enseignants utilisent la communication pour présenter des informations, impliquer et motiver les étudiants et évaluer leurs connaissances. La classification de la communication par modélisation comprend la communication graphique, l'intelligence sociale, les interfaces de composants et la communication en langage naturel, et des systèmes utilisant une ou plusieurs techniques sont présentés.

Enfin, ce chapitre présente les principes d'évaluation des STI et les cas d'utilisation de l'évaluation. L'évaluation dans les systèmes tutoriels intelligents est un processus de collecte de données pertinentes et de transformation de ces données en informations significatives utilisées pour la prise de décision.

4.2 Learning Theories

Every school teacher has their own philosophy on education, which is a set of ideas and assumptions about how to best teach (Woolf, 2010). These may be explicitly stated or implied in the teacher's approach to teaching. Educational philosophy is an important consideration for teachers, as it guides their decision-making and informs their teaching practices. On one end, some teachers see their job's main responsibility to impart information to students and then identify who has learned. They are viewed as traditionalists in teaching philosophy (Becker, 2000), the instructor is the knowledgeable source, and the student is the novice, willing to listen and learn. On the other end, there are the modern teachers who are responsible for creating experiences for students. Their philosophy is to involve learners in the activities, the approach is more dynamic and interactive. Teaching is a method to explore the curriculum. Activities proposed may be tasks, questions, and dialogues, and the constructivist teachers are more eager to use technology for inquiry-based, problem-solving forms of teaching (Woolf, 2010).

The teaching philosophy classification marks a bold difference between the two categories of teachers you can see nowadays in classrooms. The traditionalist teacher's main role is to be the explainer, the tutor who shows the students how to do the work and how to evaluate if they learned

well or not, in the same manner for everyone. The constructivist teacher acts as a facilitator, he provides opportunities and resources to the students, and they have to discover and build concepts by themselves. As the traditional approach is focused on the curriculum content and should not be impacted by the student's learning styles or motivation, the constructivist approach is well-suited for adaptation, learner's interests, and sense-making on the topics.

The constructivist teaching theory has the potential to generate new knowledge, challenge students on certain topics, and they are actively involved in the course. Students and teachers may exchange their roles from time to time; students may present and challenge the class on subjects where the teacher lacked expertise.

Human learning is a complex process and involves a variety of components and processes (Bruner, 1990). Even though there are many postulated theories in the last 50 years on human learning, there is no single teaching environment that is shown appropriately for most of the domains and people (Woolf, 2010). There are still several principles that show efficiency in human learning.

Students are learning when they are motivated, involved, engaged, and challenged in the process. The learning experiences must be authentic and relevant for them. According to the theories of psychologist Lev Vygotsky, students learn at different rates and in different ways (Vygotsky, 1978). This means that teachers must be flexible in their approach to teaching and be willing to adapt to the needs and abilities of individual students. This principle is important for ensuring that all students have the opportunity to learn and succeed. There isn't any learning method that can be applied to everyone. It has been shown that learning is more efficient if the content is customized to both learners and their current state.

The constructivist teaching philosophy has the potential to fit best the human learning principles. According to this paradigm, the approach to teaching emphasizes the idea that learners construct knowledge through real-life experiences as a cognitive activity, rather than simply learning from abstract concepts. According to Semerci and Batdi, the constructivist teaching framework includes the following five principles (Semerci & Batdi, 2015): "1) Posing problems of emerging relevance, 2) Building lessons around primary concepts, 3) Seeking and valuing learners' points of view, 4) Adapting instruction according to learners' points of view and 5) Assessing learner learning in the context of daily teaching". This approach is based on the idea that learning is an active process and that students are more likely to engage with and understand material when it is presented in a context that is meaningful and relevant to their own experiences.

ITSs are a type of intelligent system that is designed to support learning. The components of ITSs are based on certain values and beliefs about the nature of knowledge, learning, and teaching (Self &

Akhras, 2002). The architecture of an ITS typically includes a representation of the knowledge to be learned (the domain model), a system for inferring the learner's current level of knowledge (the student model), and a plan for instructional steps that will be most effective for the learner (the tutoring model). This helps the ITS provide customized and targeted support to help the learner learn and progress. ITS architecture is matching the traditional teaching philosophy. However, for the constructivist theory, Self and Akhras (2002) challenge the classic ITS architecture, the constructivist view emphasizes different values at its core and potentially requires a different architecture. Authors proposed a new approach to building ITSs where, at the core of the learning process, there are interactions between learners and tutors. They showcase their new architecture with two applications: SAMPLE – ITS to support the learning of salad-making concepts, and INCENSE – ITS for learning software engineering. Even though they briefly explain the design of the systems, no validation has been presented for any of them.

Most existing ITSs are using the traditional teaching philosophy, even though the constructivist approach has the greatest potential to enhance human learning (Woolf, 2010). Constructivist teaching is harder to be implemented in the classroom or automated systems, as it involves more time for the teacher, more resources, labor, and, in some cases, it might require hiring more personnel to support the process. Constructivist theory is suitable in technology-mediated environments, and this acts as another limitation for its adoption in classrooms.

Traditional teaching approaches can be divided into two main categories: behaviorism and cognitive sciences, which influence the paradigms under which ITSs are constructed (Alessi & Trollip, 2000). Behaviorism views learning as a process of memorization, demonstration, and imitation. In this approach, the teacher or intelligent tutor is seen as the primary source of knowledge, and learning strategies tend to focus on memory tasks and recall. Cognitive sciences, on the other hand, recognize that learning is influenced by internal factors such as motivation, attention, and metacognitive skills. This leads to a focus on active learning, transfer of learning, and comprehension, with the teacher serving as a partner and facilitator in the learning process. Many intelligent tutors are based on cognitive science principles, as this approach allows for individualization and tailoring of instruction to the learner's needs.

As explained above, constructivist teaching is harder to be enabled in both classrooms and computer-supported environments, but several methods specific to constructivism are used in actual educational systems, such as one-to-one tutoring, or case-based inquiry, as can be seen in Table 7 (Woolf, 2010).

Education based on artificial intelligence solutions uses synthetic reasoning about teaching and learning. This is a challenging task, as knowing what, when, and how to teach involves multi-disciplinary teams, covering domains including computer science, psychology, and psychology. The development of an intelligent system to fulfill a set of cognitive teaching goals is a complex process that requires expertise in the abovementioned disciplines, as can be seen in Figure 7.

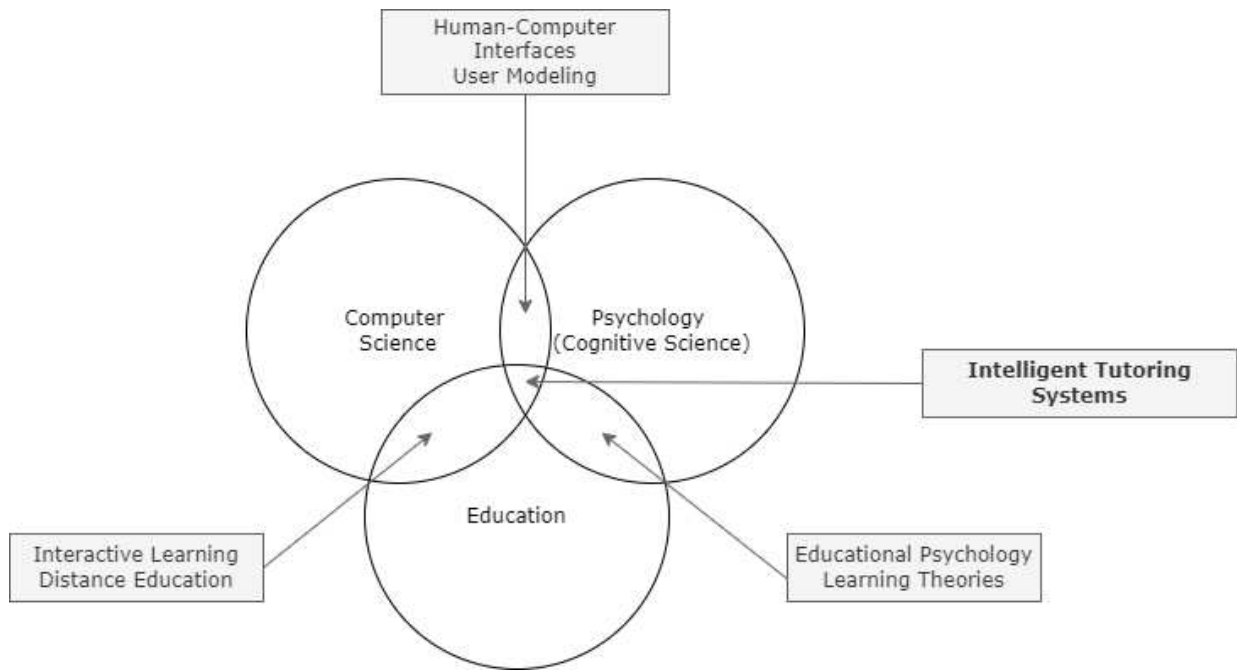


Figure 7. Disciplines involved in building Intelligent Tutoring Systems

The psychology field is concerned with tracking human learning and understanding human activity while performing learning tasks. The cognitive scientist involved reflects on how to identify the learner’s states from the beginning and the end, and which are the decisions involved to move between states. Computer science, more precisely artificial intelligence (AI), is concerned with data collection and manipulation to reproduce intelligent behavior (Shapiro 1992). Computer science also involves the usage of multimedia and the internet to facilitate the learning process. AI enables representing domain and student knowledge through inference rules. An intelligent tutor can foresee actions by using gathered experience and is able to assess the student’s knowledge before responding. Artificial intelligence and cognitive sciences are focused on answering a similar question, using different approaches. Fundamentally, both disciplines aim at achieving an understanding regarding the character of actions that are based on intelligence regardless of the material elements involved in the cognitive process (Shapiro 1992).

Table 7. Constructivist-Teaching Methods and Usages, as per Woolf (2010)

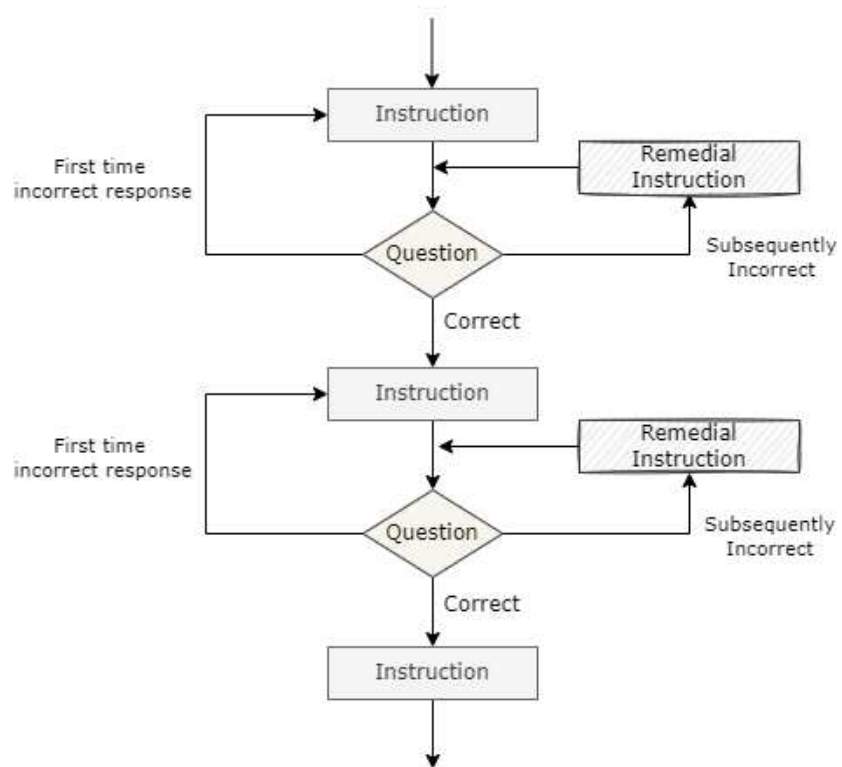
Constructivist Teaching Method	Description	Classroom Example	Intelligent Tutor Example
"One-to-One tutoring"	Dialogue between student and teacher to facilitate fixing student errors.	Students engage with teachers or older students in a dialogue where they explain their understanding.	Tutors generate problems and hints according to students' levels and tutoring objectives.
"Case-based inquiry"	Real-life scenarios are shown to students; they have to follow a path to solve the presented case.	In patients' medical symptoms, students make a hypothesis on probable diseases and try to make connections within the available data.	In computer-rich interfaces, the tutor supports the sharing and exchange of study materials, encouraging students to reason about methods and learn through trial and error.
"Apprenticeship learning"	An expert is guiding the students in the learning process.	In learning to use a piece of complex machinery, a master should engage the student in a real environment.	Computer environments can act as experts and replicate complex environments, through virtual reality, augmented reality, or mixed reality.
"Collaboration"	Teams of students cooperate in explaining the logic of reasoning about a topic.	Working together on a topic such as the reason for the extinction of dinosaurs. They study about knowledge generation, reasoning, evaluation, and revision.	Environments designed for multiple users are suitable to be used with intelligent tutors

Education is concerned with how people teach and the impact of teaching through communication, curriculum development, assessment of competences, and instilling of motivation. Educators' aim

at creating and upkeeping teaching that is simultaneously effective, affordable, accessible and efficient(Woolf, 2010). Education should focus on ways to enhance the acquisition, manipulation, and usage of knowledge, and should periodically evaluate the impact of different teaching methods on learning outcomes. This can help educators identify the most effective strategies for helping students learn and understand material, and make informed decisions about how to structure their teaching.

4.3 Modeling Knowledge and Teaching

Traditionally Computer-Assisted Instruction (CAI), initially built in the 1960 (Urban-Lurain, 1996), was able to present instructional materials in a rigid tree structure to guide the students from one knowledge component to another depending on their answers (Butz et al., 2006), as it can be seen in Figure 8. CAI systems may be effective in certain situations, but they have limitations when it comes to the diversity of learners' knowledge states and characteristics. These systems are not able to create flexible instructional plans that are tailored to the needs and profiles of individual students. As a result, they may not be as effective at helping learners as more flexible and personalized



approaches to instruction.

Figure 8. Traditional CAI Pedagogical Tree Structure

CAIs rigidity drawback has opened a new perspective for Artificial Intelligent application in education, where one of the most known candidates is the Intelligent Tutoring System (Butz et al.,

2006). ITSs can provide flexible and personalized instructions similar to one-to-one tutoring. The system is also able to provide individual feedback to learners.

One of the key concepts that distinguish an ITS from a CAI is ITS's ability to keep the learning and reasoning model of the student in a dynamical manner, which evolves while the system is used. As stated by Shute and Psozka (Shute & Psozka, 1995), an ITS should be able perform the following actions:

- Precisely assess the level of knowledge through interaction based on principles;
- Design the journey and adaptively adjust the learning path;
- Help the student become aware of their evolution process.

The modeling assessment techniques and intelligent adjustments are what distinguishes an ITS from a CAI. This is accomplished mainly using artificial intelligence techniques. Intelligent tutors, like human teachers, must use various teaching strategies to interact with students who have different abilities and learning styles. In order to do this effectively, tutors must have a wealth of knowledge that can be divided into four main categories: student knowledge, domain knowledge, tutoring knowledge, and communication knowledge. Some intelligent tutors use combinations of these components. The modeling of knowledge for each of these components will be discussed in more detail later.

4.3.1 Domain and Student Modeling

Learner modeling is achieved by construction for various objectives, including for the purpose of detecting solution routes, assessing capabilities related to finding solutions, and identifying potential errors made by the learner. (R. Sottolare, Graesser, Hu, & Holden, 2013). The content included in the learning model can be domain-specific or domain-independent. Domain-specific information exposes the student's current knowledge level or mastery of a subject area (R. Sottolare et al., 2013). Domain-independent information includes student learning particularities, such as interests, demographics, past performance, and domain-general competencies, as well as psychological measures, affective and cognitive dimensions, and personal beliefs. These models can be used to understand the learner's needs and tailor instruction accordingly.

Common techniques for generating learner models include Bayesian networks, belief networks, case-based reasoning (CBR), and expectation-maximization. The learner models can be classified by their performing function (R. Sottolare et al., 2013). During the learning cycle, all six functions serve different purposes. Learning strategies are devised, content is elaborated based on predictions about the learning needs, knowledge diagnosis is followed by necessary corrections, and overarching

evaluations enhance the interaction with the system and the awareness of the student by quantifying evolution.

The main reason for student modeling is to provide support for the intelligent tutor in order to possess sufficient information about individual learners in order to act constructively and efficiently, and to engage the student's interests. (Woolf, 2010). This is similar to the approach of a master teacher, who adapts their instructional strategy based on various factors such as affective characteristics such as body language, facial aspects, and paraverbal signals. By building a student model, the tutor can better understand the needs and preferences of each learner and tailor instruction accordingly.

The foundation of student model construction relies on the way that a tutor reasons on a subject (domain models), shared representations of learner understanding and misunderstandings (overlay models and bug libraries), the information provided by students (bandwidth), and the ability to help the learner assess their own level of understanding (open student models). These concepts are important for building effective student models that can help the tutor understand and support the learning needs of individual students (Woolf, 2010).

Many ITSs developments consider the student model to be a structure mapped onto or a delimited subpart of the domain model (Ma et al., 2014). As the ITS aim to teach the domain or a part of the domain, initial work, before representing student knowledge, should be the definition of the domain model.

In a traditional ITS, the domain knowledge representation has been implemented through a) black-box models, where reasoning is not clearly explained, but the solutions are accurate; and b) glass-box models, where the reasoning is explained step by step (Polson & Richardson, 1988).

There are two types of knowledge in both models: a) declarative, which is conceptual information; and b) procedural, for action sequences and problem-solving procedures. The goal of these representations is to make sure that structured knowledge and proper learning sequences are accessible for the tutor module (Zouaq & Nkambou, 2010). Several representation formalisms have been proposed and used traditionally in ITS, such as simple rules, case-based reasoning, fuzzy logic, concept maps, topic maps, or conceptual graphs (Zouaq & Nkambou, 2010).

The overlay model for designing student knowledge is built on top of the domain model. Rules and weights for each expert step are most of the times annotated onto the knowledge representation. In this way, the progress and achievement of the student over a knowledge component is easy to assess. Misconceptions can be represented as additions to the overlay model. This method has a setback

due to the fact that the student will have knowledge only specific to the domain, while the learner characteristics are more complex (Chi , Feltovich , & Glaser 1981).

Bug library modeling refers to representing the student model within a predefined framework of misconceptions (VanLehn, 1988). This approach implies manually assessing the likely bugs that the students may arise; the predefined library of misconceptions needs to be as complete as possible. Bug library modeling is a limited approach as it is not productive and not suitable for complex, non-procedural domains.

Bandwidth refers to all the information the tutor can track about a student. Open user models allow students to inspect and control user models and be engaged in their development and updates (Bull & Mabbott, 2006). Learners are more engaged in the process when comparing their levels of knowledge with their peers' or tutors' expectations. Even with learners' access to the student model, the open user models are not necessarily more accurate and do not offer the opportunity to personalize learning.

Student models represent many types of knowledge, which can be represented in a variety of ways, as can be seen in Table 8 (Woolf, 2010). However, with the advancement of the Semantic Web, the development of the Educational Semantic Web in particular (Aroyo & Dicheva, 2004), and the lack of uniformity between the approaches used for knowledge representation, ontologies are now the key to increasing the speed of building ITSs. Ontologies enable the reuse of existing knowledge from similar systems, standardization of knowledge representations, and support bridging ITSs and eLearning resources (Nkambou, 2010).

A domain ontology is a strong alternative for knowledge representation when building ITSs for their standard formalism, ease of reuse of other ontologies, and modularity. System designers must integrate different ontologies to enforce the reuse and interconnection of various relevant resources.

Table 8. Knowledge Representation in Student Models, as per (Woolf, 2010)

Knowledge Category	Knowledge Type	Representation
"Topics"	"Concepts, facts, procedures; rules, skills, abilities, goals, plans, and tasks; declarative knowledge about objects and events"	"Overlay plan of facts and procedures, Bayesian belief networks, declarative knowledge"

"Misconceptions and bugs"	"Well-understood errors, «buggy knowledge», missing knowledge"	"Bug library, bug parts library, mal-rules"
"Student affect"	"Engagement, boredom, frustration, level of concentration"	"Reinforcement learning, Bayesian belief network"
"Student experience"	"Student history, student attitude; discourse, plans, goals, context of the user"	"Recover all statements made by students; identify patterns of student actions"
"Stereotypes"	"General knowledge of student's ability and characteristics; initial model of the student"	"Build several default models for different students; store most likely values"

An ontology is a shared vocabulary and representation of knowledge used to model a domain; ontologies define explicit descriptions of concepts and their relations and integrate computer-processable semantics for data on the Web (Fensel, 2001). In an ontology, all concepts must be explicitly defined in a machine-understandable format. As part of the Semantic Web domain, ontologies are well-fitted to support data integration, while concurrently organizing knowledge. Combining data from different ontologies may lead to the identification of new relationships between concepts while helping to eliminate or reduce the ambiguity between terms that belong to distinct sets of data (Noy & McGuinness, 2001). One challenging task, however, is to match and record the relationships between entities in the linked ontologies, considering their continuous evolution.

A Modeling Knowledge using Ontologies in Intelligent Tutoring Systems

Ontologies are commonly used in a wide range of applications, including organizing information within a database, the search engine of a peer-to-peer system, websites where customers need efficiently to search for products, and semantic web services in general (Euzenat & Shvaiko, 2007). An ontology usually consists of a vocabulary that describes a particular domain, along with a

specification of the terminology meaning. The Resource Description Framework (RDF)¹⁰ is a standard describing and organizing online digital resources and the related metadata with the help of a grammar-based notation and a storage efficient data format. It allows for the manipulation and querying of data using SPARQL in knowledge representation systems. Ontologies are useful for organizing and representing knowledge in a structured and systematic way.

In data representation systems, such as RDF graphs, data is organized into triples consisting of a subject, predicate, and object. The predicate represents the relationship of the subject (a node in the graph) and the object (a value or another node from the graph). The Ontology Web Language (OWL) (McGuinness & Van Harmelen, 2004) is a standard for defining ontologies, which provides the possibility of creating specific class hierarchies, domains, relationships, and constraints and axioms. Protégé (Noy et al., 2003) is a commonly used software for constructing large electronic knowledge bases, which provides tools for a developer to model and modify a domain ontology directly. Ontologies can be of use for representing knowledge in ITSs, as they provide a structured and systematic way to organize and represent information.

The GET-BITS model was built as a generic shell for developing ITSs, using at its core an object-oriented model, class hierarchies, and class design principles for knowledge representation (Devedzic, Jerinic, & Radovic, 2000). The model defines five levels of abstraction, ranging from top-level (i.e., Integration, where Domain knowledge represents the curriculum, and the Student module stores the progress in group learning) to the bottom (i.e., Primitives, where Domain knowledge for a computer science topic may be represented as a logical expression or a clause, and the Student model considers learning speed, knowledge level, or the level of concentration). Central classes described in the GET-BITS model include the Lesson, the Topic, the PQ (problems and questions for the student), and the Explanation class, all of them linked to a Frame from the learning sequence., as can be seen in Figure 9.

¹⁰ <https://www.w3.org/RDF>

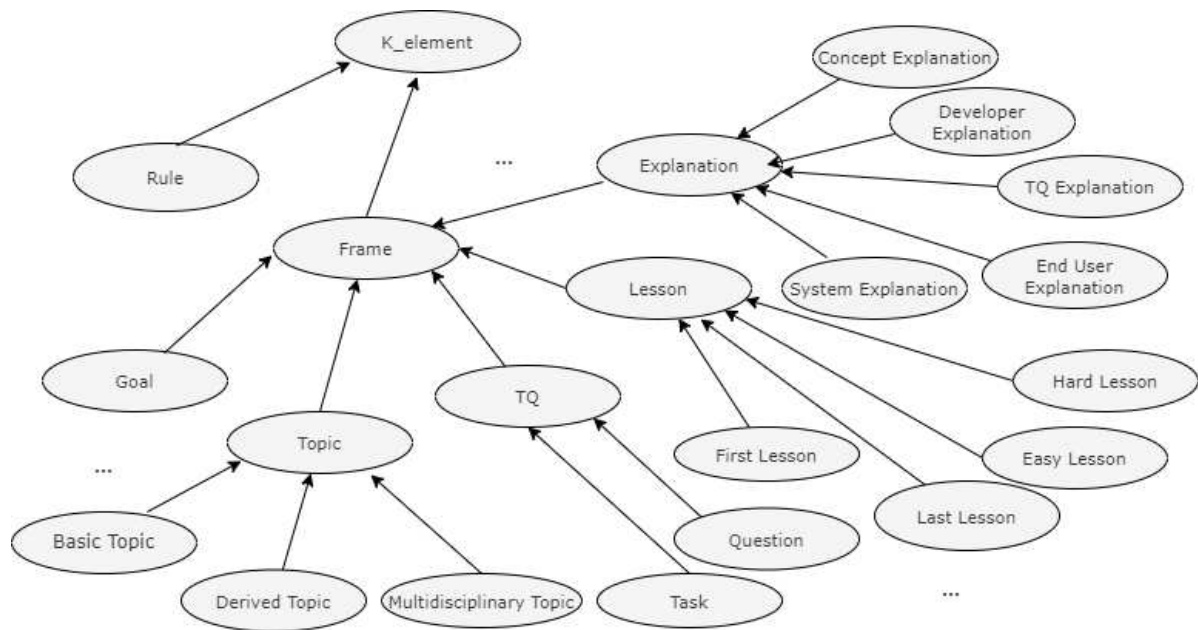


Figure 9. GET-BITS model - Main classes used for knowledge representation (Devedzic et al., 2000)

However, Ali Ahmed and Kovacs (Ali Ahmed & Kovacs, 2020) have presented many recent approaches which integrate ontologies into Intelligent Tutoring Systems. They build a tutor for learning Java programming language using Protégé for building the ontology and Python for the integration with the knowledge repository.

An ontology-driven method for an ITS was developed by Dermeval et al. (2019). The goal was to develop a gamification module in an ITS by connecting concepts in a standard and formalized manner. The knowledge base was structured in ontologies and covered concepts such as gamification, domain, student, and tutoring.

The link between ITS concepts and gamification facilitated an automation of the reasoning process involving system managed information, enabling their interoperability and defining good practices for creating a gamified ITS. Two project ontologies were related to domain knowledge and were divided into core and specialized concepts that described gamification. A third one was a Tutoring Ontology that connected ITS concepts from an ontology (i.e., Gamified ITS, Domain, Student, and Pedagogical models) with gamification concepts imported from the two previously mentioned domain ontologies.

The connection relied on technologies such as RDF/OWL vocabularies (e.g., FOAF), and modeling in Protégé. Methontology (Fernández-López, Gómez-Pérez, & Juristo, 1997), a methodology for building ontologies based on the IEEE standard criteria, was considered throughout the entire ontology life-cycle process.

Panagiotopoulos et al. (Panagiotopoulos, Kalou, Pierrakeas, & Kameas, 2012) designed a student model for an ITS and enhanced it with semantics using an ontology. Their Student model integrated student characteristics and progress, while their ITS provided personalized educational content based on student learning and academic particularities. The study targeted facilitating the communication between intelligent agents through a domain-independent vocabulary while modeling student profiles. Their ontology was composed of two main taxonomies, containing records about student academic and personal information.

Moreover, the ontology captured, besides basic characteristics of students, details on their learning style using the Felder-Silverman model (Graf, Kinshuk, & Leo, 2006), an approach that combines stereotype and overlay techniques. The ontology was developed in Protégé using Ontology Development 101 (OD101) methodology and adopted OWL as a formal representation language (Noy & McGuinness, 2001). In addition, the authors developed inference mechanisms as a set of rules using Semantic Web Rule Language (SWRL) for the automated classification of the students into stereotypic profiles.

As ITSs were mainly built to develop cognitive capacities, no previous work was found in terms of ontologies used in ITS for psychomotor development. However, other ontologies like the Sports Domain ontology (Ramkumar & Poorna, 2017) were previously developed as broad coverage of the sports domain using Protégé and following the Ontology Development 101 framework. Their conceptualization leveraged five sports – football, rugby, cricket, athletics, and tennis. The work presented 840 classes with their corresponding properties, relations, and axioms. The development process included the instantiation of the classes in the domain, and the evaluation was performed using *ref* – a Semantic Web framework written in Java, with features for parsing queries, model representation, or data visualizations.

Diaz-Rodriguez et al. (Diaz-Rodriguez, Wikstrom, Lilius, Cuellar, & Flores, 2013) developed an ontology in the psychomotor domain by using the Kinect 3D depth sensors for activity recognition, semantic reasoning, and semantic modeling in movement and human interaction. Their goal was to distinguish between human movement, human-object interaction, and human-computer interaction. The research team proposed as future work the implementation of a fuzzy approach, as the results obtained with the Kinect devices generate inaccurate data for their feature modeling experiments.

My state-of-the-art survey did not identify ITS initiatives to develop bio-motor abilities, such as strength skills. Existing ontologies in the sports domain provide the foundation to develop digital solutions to recognize sports activities or support decision-making based on data collected during

competitions. However, existing work does not provide the knowledge required to support the development of bio-motor abilities.

4.3.2 Teaching Modeling

A significant challenge in the development of an intelligent tutoring system (ITS) is how to effectively model the tutoring module (Zouaq & Nkambou, 2010), what kinds of knowledge representations are available, and what types of knowledge acquisition techniques can be used. The tutor must have a clear representation of the domain knowledge that is the focus of the learning goal. This is an important research issue in the field of ITS, as the ability to accurately represent and use domain knowledge is essential for effective tutoring and student learning

Teaching knowledge is essential for a tutor, as it allows them to make informed decisions about when to intervene based on the student's presumed knowledge, learning style, emotions, and current state(Woolf, 2010). Teaching modeling involves selecting strategies for interventions, customizing responses, and engaging students. Tutoring strategies can be classified based on several criteria(Woolf, 2010):

- Observations of human teachers (such as apprenticeship training, problem-solving, tutorial dialogue, and collaborative learning);
- Learning theories (such as ACT-R, the zone of proximal development, and social interaction);
- Technology-driven approaches (such as pedagogical agents and virtual reality).

The use of technology has enabled new approaches to education, such as the flipped classroom model. In this model, homework and lectures are reversed, with students watching pre-recorded video lectures before attending class and using class time for problem solving, inquiry, seminars, and educational games. An intelligent tutoring system can be used to support the flipped classroom model. Studies have shown that the use of a flipped classroom intelligent tutor can have a positive impact on student satisfaction and learning outcomes (Mohamed & Lamia, 2018). However, there are limitations to this approach, including a small number of qualitative video lessons and a lack of continuous measurement of the differences between traditional and flipped classroom teaching methods. Additionally, the results of this approach may not be generalizable to a wider population due to the limited number and similarity of participants.

Problem-solving tutors are commonly used in traditional teaching. The student knowledge level is estimated through multi-step problems. Such tutors are suitable for quantitative domains (Woolf, 2010): mathematics, where students' solutions are analyzed by using equations, graphs, or plot

points; physics, where students learn to describe phenomenon using equations and indicate forces through vectors; or computer science, where students have to think algorithmically and input procedural code. The issue of this approach is underlined by the cognitive load theory – when learning through problem-solving, students tend to be focused on resolving the problem, mainly using relatively weak search strategies and limiting comprehension (Sweller & Chandler 1994).

The process of building problem-solving tutors must involve a bug library, through which the tutor can match student mismatching responses to related incorrect knowledge. Bugs may indicate an incorrect thinking pattern or limitations in acquiring the grammar patterns of the programming language. From our findings, many tutors have been built using problem-solving teaching strategies, mainly in the exact sciences fields. In mathematics, the Wayang Outpost tutor has been developed, within an animated learning environment, where students play and learn geometry, and receive constant hints, and personalized learning sequences (Arroyo, Beal, Murray, Wallis, & Woolf, 2004). The authors demonstrated that macro-adapting the hints to students' cognitive skills can yield higher learning outcomes.

Another problem-solving tutor has been used in learning classical physics – Andes (Schulze et al., 2000). The tutor authoring environment is a system designed to help authors create intelligent tutoring systems. It is constructed with 600 rules that cover all the equations needed to solve each exercise, including a graph of all reasonable solution paths. The flexibility allowed the students to determine their solution path without being forced to solve it in one only way. The experiments conducted showed an increase in performance by 3% for the students using the Andes than students learning in a traditional environment.

FITS (Flowchart-based ITS) is an intelligent tutoring system designed to help students improve their C++ programming skills through problem-solving (D. Hooshyar, Ahmad, Yousefi, & Yusop, 2015). It integrates an interactive environment that guides users through the development of flowcharts for imperative and practical programming problems, using Bayesian networks to personalize the learning process for each student. Studies have shown that FITS is effective in improving learning outcomes and can be particularly helpful for students with lower levels of prior knowledge. It is an example of an ITS that actively engages users in the learning process and can be particularly effective in helping them develop practical skills.

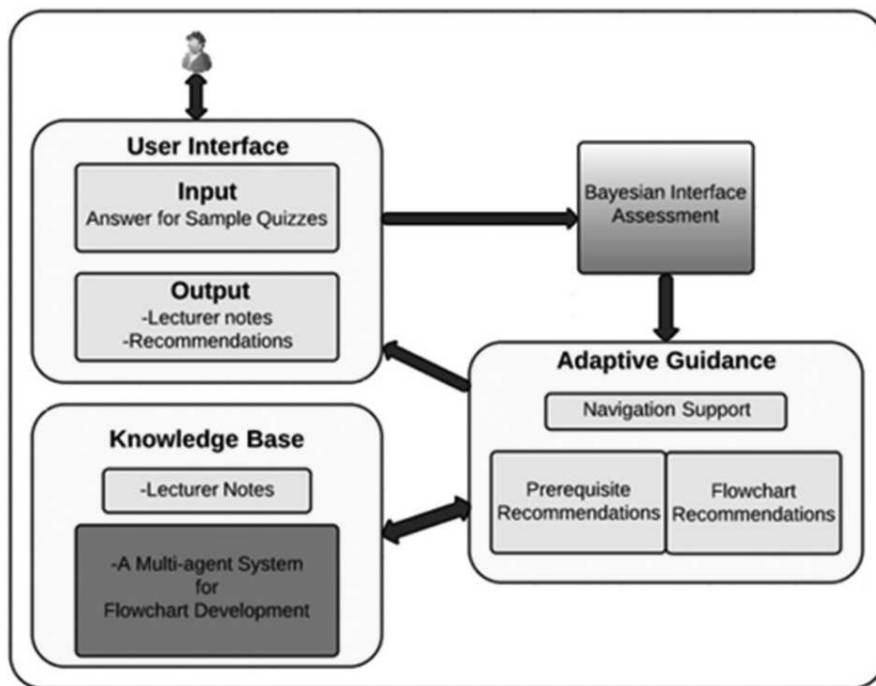


Figure 10. FITS Problem Solving Tutor Architecture (D. Hooshyar et al., 2015)

Based on the human learning theories, the tutor's classification relates to different aspects. No learning theory may be applied for all situations to all learners. Human learning specialists have a pragmatic viewpoint on the domain being taught, the nature of the learning, and the learners' level. If the domain involves a higher level of processing, a common learning theory used is based on the constructivism rules. For a domain with prescriptive solutions, such as algebra, a systematic learning approach might be useful. In cases where the learning involves the integration of multiple tasks, such as in the management of a software project, a learning theory based on social learning can be efficient.

The cognitive learning theories are very common in modeling knowledge in intelligent tutoring systems (Woolf, 2010). The specialists are responsible to take over a task, break it down into smaller pieces, and use those bits to develop instructions to teach students, from basics to advanced principles. Some of the key principles in cognitive learning are (Mergel, 1998): creating meaningful learning information, serial position effect (the knowledge items that are placed first and last in a list are effortlessly remembered), smart repetition strategy to improve memorization, effects of prior learning and interference when learning.

A cognitive learning theory used in modeling tutors is ACT-R (Adaptive Control of Thought), which states that human cognition is created by the interaction of knowledge that is procedural and declarative. In respect to knowledge, the term declarative refers to factual information, is modular, based on working memory, and has a limited size. Argumentations like " $4 * 8 = 32$ ", or "Every if

query receives a boolean statement" are examples of declarative knowledge. On the other side, procedural knowledge refers to rules on step-by-step instructions for solving a problem. The procedural type is superior to the declarative one since it involves abstract effective algorithms. Argumentations like "learn how to transform from base-10 to base-2" or "learn to programmatically compute certain Fibonacci number" is known as procedural knowledge. ACT Tutors are also called model-tracing tutors and some recent tutors have been built in mathematics (King, Kelvin, Warnars, Nordin, & Utomo, 2021), or computer science (Marouf, Abu Yousef, Mukhaimer, & Abu-Naser, 2018). Also, during the last 15 years, model-tracing has been intensively used for authoring ITSs, as Dermeval et al. shown in a recent literature review (Dermeval, Paiva, Bittencourt, Vassileva, & Borges, 2018). An authoring tool is a software program that enables teachers, designers, and developers to create learning content (lessons and courses using text, media, or another type of interaction). The users creating such content do not need any technical programming expertise to use the authoring tool. Such tools make the implementation of an ITS faster, optimize production workload, and create an abundance and diversity of tutors.

The extensive literature review (Dermeval et al., 2018) has shown that the model-tracing/cognitive tutor approach is the most commonly used approach in the development of intelligent tutoring systems (ITSs), accounting for 21.21% of the systems studied. Other commonly used approaches include example-tracing (18.18%), content and problem-based (12.12%), and dialog-based (9.09%). The remaining approaches accounted for less than 6% of the systems studied. The review also found that 39.39% of the ITSs studied were presented as illustrative scenarios, 27.27% were controlled experiments, 15.15% were case studies, and 3.03% were surveys. The remaining 15.15% of the systems were not clearly described or were non-specific.

In the majority of cases, intelligent tutoring systems utilized authoring tools for the creation of tutoring models (81.82%, 27 studies). These were followed by the use of authoring tools for domain modeling (75/75%, 25 studies), student modeling (18.18%, 6 studies), and communication modeling (15.15%, 5 studies). It's important to note that in some cases, a single study utilized authoring tools for more than one component of the intelligent tutoring system, which is why the total percentage exceeds 100%. The use of authoring tools across all components is illustrated in Figure 11. The authors refer to the tutoring model as a pedagogical model and the communication model as the interface model in their literature review.

The example-tracing tutors are responsible to interpret and assess the student's behavior regarding generalized examples of problem-solving behavior (Aleven, McLaren, Sewall, & Koedinger, 2009). This paradigm has been proposed by Aleven 13 years ago and it has been adopted very fast; 18% of the selected studies in the literature review have been using example tracing as an authoring strategy.

Building a cognitive model through demonstration rather than by rule model programming is facilitated for the domain experts by this specific strategy. The outcomes have a positive impact by reducing the development costs overall.

Example-tracing tutors can provide sophisticated tutoring behaviors, maintaining a set of distinct student behavior interpretations and recognizing multiple student strategies. Alevan et al. (2009) proposed also a framework, Cognitive Tutor Authoring Tools (CTAT), where such tutors can be created without programming. Designers can drag-and-drop techniques and then implement problem-solving behaviors which are intended to be tutored. The behaviors designed are represented in a behavior graph, which can be seen, edited and generalized.

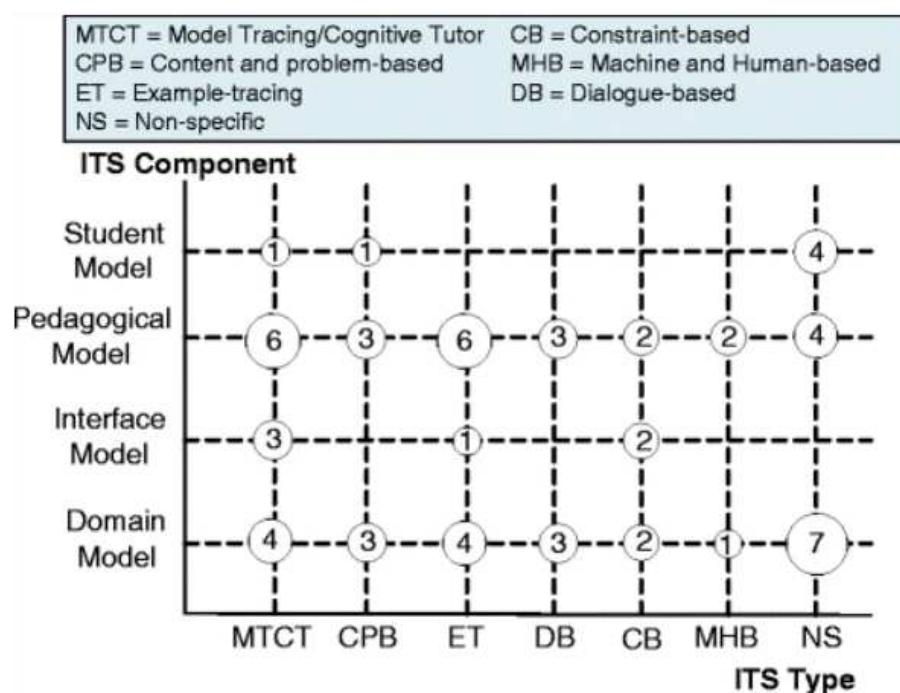


Figure 11. Authoring Tools used for modeling ITS components (Dermeval et al., 2018)

Tutorial dialogue or dialogue-based tutoring relies on NLP techniques to offer tutoring to the learners that resembles human to human interaction. The aim is to create systems that can reason and conversate as flexibly as humans, and understand remarks, discussions, and student feedback. These systems rely heavily on physical machines or human intelligence as supplementary support while tutoring (Woolf, 2010). More details on how such tutors are built will be presented in 4.3.3. Machine and human-based tutors are systems that use a combination of physical machines and human intelligence to aid in the tutoring process. These types of tutors rely heavily on both elements to effectively teach and support learning.

Dermeval’s review shows that there are several categories of features that were used by ITS designers to help the authoring process. A tutor may have more than one feature. 21 categories have been

identified. Based on the ITS components where they were implemented, the detailed view can be seen in Table 9. There were some studies for each no special feature has been identified, they were marked in the Not Applicable category.

Table 9. Features used for aiding the authoring process in ITS Models, as per (Dermeval et al., 2018)

ITS Model	Feature	Number of studies	Percentage of studies
"Student Model"	"Define students' stereotypes"	2	6.06 %
	"Authoring based on learning styles"	1	3.03 %
	"Reuse of students' profiles"	1	3.03 %
"Domain Model"	"Define problem solutions"	6	18.18 %
	"Authoring by demonstration"	5	15.15 %
	"Automatic domain model generation"	4	12.12 %
	"Define hints"	3	9.09
	"Reuse of learning content/domain model"	2	6.06 %
	"Human computation"	1	3.03 %
"Pedagogical Model"	"Define / Give feedback"	9	27.27 %
	"Define behavior graphs"	3	9.09 %
	"Make assignments"	3	9.09 %
	"Define cognitive model"	2	6.06 %
	"Define collaboration scripts"	1	3.03 %

"Interface Model"	"Drag and drop interface authoring"	2	6.06 %
"General"	"View learners' statistics"	4	12.12 %
	"Mobile authoring"	2	6.06 %
	"Reuse / Export Tutor design"	2	6.06 %
	"Create class lists"	1	3.03 %
	"Not applicable"	4	12.12 %

Natural language (NL) intelligent tutoring systems (ITS) aim to replicate human-like communication and understanding in order to support learning. These systems consider the importance of interpersonal activities in the process of knowledge acquisition, as studies have demonstrated that it significantly enhances the developing of the cognitive function. According to Vygotsky's theory, a combination of interpersonal activities and the Zone of Proximal Development (ZPD) is effective in teaching (Vygotsky, 1978). The idea is that learners can acquire skills from experts through social interaction, and that knowledge acquisition is driven by activities practiced in a social context.

The Zone of Proximal Development is the essential ingredient in effective instructions (Woolf, 2010). It implies the amount and complexity of development of a student attained through interpersonal relationships which is superior to individual solitary learning. An example of ZPD implementation in cognitive tutors was performed by (Clement et al., 2015), for a tutor which teaches numbers decomposition, and it can be seen in Figure 12. The implementation of ZPD depends on full social interaction, as it requires collaboration or support from another learner or the tutor. Based on these considerations, building a ZPD strategy is more suitable for apprenticeship tutors but, from my findings, so far it has been used only in cognitive skills development.

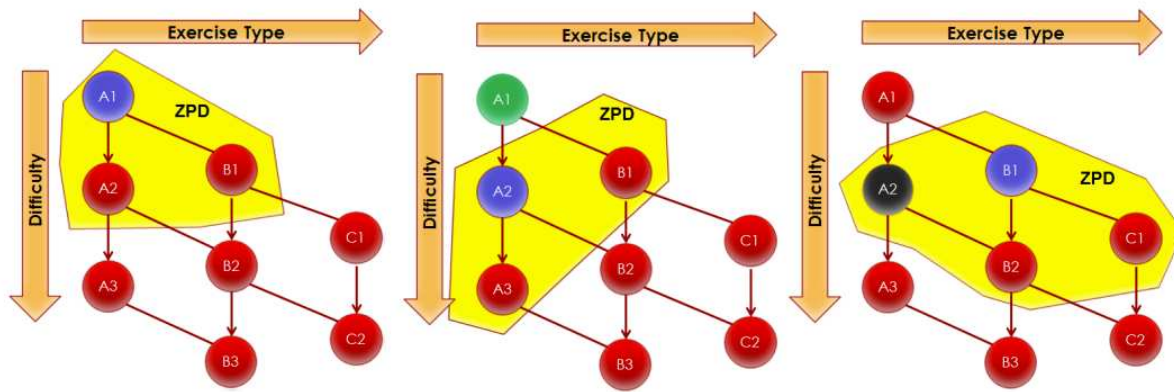


Figure 12. Zone of proximal development evolution according to observed results. (Clement et al., 2015)

The apprenticeship teaching strategy involves a student learning through observation and imitation of an expert, rather than receiving explicit instruction. This strategy is modeled after the way human tutors teach, and involves the student acquiring and developing their own cognitive tools for learning, both in and out of school (Brown, Collins, & Duguid 1989). The apprenticeship tutor is responsible to monitor student performance, reflecting on students' approaches, can provide advice on demand, and the path to solutions should be modeled through multiple paths. The tutors usually encourage the practice and provide advice to the learners to update their reflection on their belief structures (Shute & Psotka, 1995). Apprenticeship training is common in learning how to play an instrument, training for athleticization, or learning how to drive.

The purpose of apprenticeship training is to allow students to acquire strong mental models through an accurate representation of the learning environment (Woolf, 2010). While learning, students must mimic the actions required and the expert should monitor and provide feedback to actively diagnose and fix student misconceptions. The process of building an apprenticeship intelligent tutor involves strong student modeling, efficient communication modeling between tutor and learner, which is later detailed in 4.3.3, and expert modeling on which advice to present.

The apprenticeship teaching strategy is a way of teaching that involves a student learning through observation and imitation of an expert. The aim of the discussed approach is to aid the learners in developing mental models that are robust and accurate through realistic replicas of learning conditions. In this method, process models are utilized in tracking a learner's progress with the help of different strategies and compare them to an internal model, which can help identify misconceptions. One example of an ITS that used this approach is SOPHIE (Sophisticated Instructional Environment), which was developed in the 1980s and used real-time natural language techniques to interact with students and help them develop troubleshooting skills (Brown et al., 1982).

Another example of an apprenticeship tutor is Sherlock, with an initial version in 1992 (Lajoie & Lesgold, 1992) and an updated version in Sherlock 2 (Katz, 1998). The tutor's goal was to learn electronics through practice. It supported students' solutions when diagnosing problems by constantly providing feedback and hints, in video, or written format when they were stuck. The new generation – Sherlock 2 encapsulated more knowledge and more reflective capability which made its evaluation to be remarkably successful, even though its design was rigid (Lajoie & Lesgold, 1992).

TUMA intelligent tutor (Hodaie et al., 2018) propose the apprenticeship tutoring approach for their manual-procedure activities learning system, even though they do not detail the implementation strategies, underlying that both cognitive and psychomotor skills, needs to be developed through practice and tutor real-time feedback.

MacLellan and Koedinger (2020) argue that the current practices for developing tutors involve designers working with domain experts to select useful content and instructional approaches for a particular domain. However, it can be difficult to access both domain experts and students for task analysis and to translate existing learning theories into component models for tutor design. As a result, designers often depend on personal experience and insights in the design process, which can lead to pitfalls in the tutor design process. The authors propose the use of a generic Apprentice Learner Architecture as a way to support efficient tutor authoring and to make the process more domain-general.

MacLellan and Koedinger's Apprentice Learner Architecture (MacLellan & Koedinger, 2020), which is presented in Figure 13, is a framework that includes both the learner and the tutor. It consists of knowledge structures (represented by blue boxes), performance components (represented by yellow diamonds), and learning components (represented by green circles). The Selection, Action, and Input (SAI) components, depicted in the figure as blue boxes, indicate the contents of one step. This architecture is designed to support efficient tutor authoring and is intended to be domain-general.

The Apprentice Learner Architecture is a proposed framework for designing intelligent tutors that incorporates the principles of apprenticeship teaching as indicated in Figure 13. It includes "four knowledge structures (Working Memory, Relational Knowledge, Overly-General Operators, and Skills), three performance components (Relational Inference, How-Search, and Skill Execution), and four learning mechanisms (How-, Where-, When-, and Which- learning)". The architecture is abstract in its implementation, but is guided by the principles of apprenticeship teaching, which focus on helping students construct resilient mind maps through practical replicas of studying conditions. The framework includes steps for matching skills with problems, generating a sequence of mental operations to demonstrate a skill, inducing relational heuristic conditions to constrain new

skills for later use, classifying new skills based on certain conditions, and updating the usefulness of learned skills.

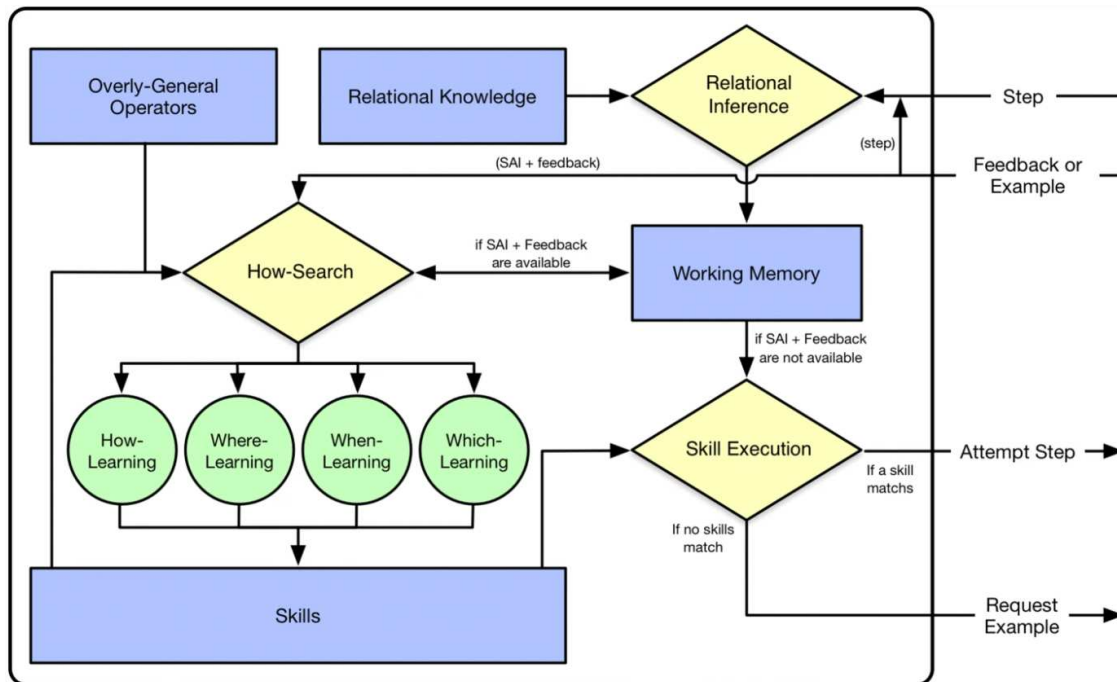


Figure 13. Apprentice Learner Architecture, as per (MacLellan & Koedinger, 2020)

Based on the teaching strategies and the skills developed in the existing tutoring systems built using this approach, apprenticeship teaching is one of the strongest candidates for developing psychomotor tutors. The previous methods introduced are more suitable for cognitive learning.

4.3.3 Communication Modeling

An intelligent tutor has limited value if the communication component does not implement efficient strategies. Communication modeling is a comprehensive process. Tutor-learner interaction should be simple, clear, and efficient. Several techniques have been implemented for communication modeling, including animated tutors, virtual reality, visual recognition of emotion, and natural language processing.

Teachers use communication to present information, engage and motivate students, and assess their knowledge. In traditional schools, teachers are typically viewed as the source of information, while students are passive recipients. An alternative approach to education emphasizes that knowledge is generated when students actively construct their understanding and organize their knowledge (Woolf, 2010).

This emphasis on student understanding and the use of knowledge, as well as the idea of challenging and inquiring, is supported by the theory of modern education that emphasizes the importance of

questioning in the learning process. According to Socrates, all learning occurs through the asking of questions, allowing students to discover solutions for themselves. This approach to education places a focus on the student's role in constructing their own understanding, rather than simply receiving information from the teacher.

In classes, teachers can depict students' reactions to certain pedagogical actions. They can see and make assumptions based on their voice inflections, eye, and body movements. Master teachers can follow students to see who is learning, by watching who is taking notes, asking questions, and also students who are potentially not listening. One-on-one communication with students has the potential to be more engaging and personalized, as a teaching strategy that works well for one student may not be as effective for another. However, individual or small-group communication between students and teachers can be challenging to implement due to the time and resources required. While this type of communication may be more effective, it can relatively be costly in terms of time and money to implement in schools.

In a recent study (Taub et al., 2019) , the emotions of students engaging in self-regulated learning (SRL) with an Intelligent Tutoring System (ITS) were examined. The authors conducted a review of literature on the impact of emotions on the learning process while interacting with Adaptive Learning Technologies (ALT), including ITSs. The learning process is typically divided into four phases: task definition, path towards objectives, acting out (e.g., strategies used by learners), and adjustments (e.g., modifying previously chosen techniques to attain the objective). Every step involves cognitive strategies and metacognitive processes, and students need guidance to monitor, search for, and assemble information at each phase. While learning environments cannot fully replace human-to-human interaction, they may provide a motivational boost and can reduce anxiety or increase student involvement in certain situations.

The literature review done by Taub et al. (Taub et al., 2019) showed different patterns. Students who created self-driven prompts exhibited increased immediate and delayed learning outcomes than the others. Also, students who used systematic reading, note-taking approaches, and informed guess-and-check approaches showed higher performance overall. Studies also examined the students' emotions while learning and proved that the positive emotional clusters scored the highest, while negative activating, such as frustration, and negative deactivating, such as boredom, showed lower learning outcomes. One study assessed scores of six emotions ("joy, surprise, anger, contempt, confusion, and frustration") across participants through facial recognition software which tracked them while learning, and it showed that emotions are linked with the learning outcome. And so, tracking emotions while learning through intelligent systems is a critical task.

An efficient tutor communication module makes learners feel authentic, and social, involving the reciprocal exchange of information with the system. ITSs built in the past 30 years can mimic many human communicative strategies, implemented based on human teachers' observations, such as speaking, critiquing, or role-playing, or based on other technological opportunities, such as virtual reality environments, or animated pedagogical agents (Woolf, 2010). Intelligent tutors are able to communicate with learners using natural language techniques, such as spoken or written explanations. They can also analyze students' explanations and assess their progress through techniques like automatic essay analysis and scoring, which analyze linguistic and textual features. (Dascalu, Westera, Ruseti, Trausan-Matu, & Kurvers, 2017). This allows the tutors to provide feedback and engage in a dialogue with the learner to support their learning. More advanced tutors can track and interpret students' affect, such as emotion, the focus of attention, or motivation. Such tutors require a more advanced setup – students need to be recorded while learning and videos should be processed in real-time to automatically detect emotions.

Efficient communication modeling must comply with a set of human tutor-like standards – it must be able to track students' responses, which can be manual input of text or speech, and also the affective characteristics – motivation and emotions. A great interface may overwhelm other tutor modules. On the other end, a poor communication interface may have a negative impact on the learning outcomes and may limit the value of the other modeling components – student, domain, and tutoring. The classification of modeling communication made by Woolf et al (Woolf, 2010) includes 4 main categories: graphic communication, natural language communication, component interfaces, and social intelligence.

Graphic communication includes animated pedagogical agents, synthetic humans, and virtual reality environments. The animated pedagogical agents may have the visual aspect of a cartoon character, but they act independently to attain teaching goals and interact with the learners. Reasoning on the actions, receive input, process, and compute output in the environment is achieved through AI-based methods.

According to a recent review of pedagogical agents in tutoring systems (Martha & Santoso, 2019), these virtual characters can come in different forms, including text, voice, 2D, 3D, and human. They can support the learning process and use various instructional strategies in interactive learning environments and can be designed as single or multi-agents. The review found that the combination of pedagogical agent design and role design can have a positive impact on learning outcomes and student behavior. Pedagogical roles may include:

- "Instructor / Navigator / Guidance (ING)"

- "Collaboration Assistant (CA)"
- "Expert / Mentor (EM)"
- "Metacognitive Suggestion (MS)"
- "Self-Report Measures (SRM)"
- "Hint / Feedback / Prompt (HFP)"

Virtual reality has the potential to immerse the learner in an environment where he can interact with a pedagogical agent. The benefit of virtual reality, from a tutor's perspective, is that this technology can assess real-time the learner's physical actions, such as body movements, as well as speech and text. In virtual reality, students have the closest experience to a traditional tutor, he can interact and collaborate on tasks, as in real life. Virtual reality communication modeling has the potential to be a great candidate for modeling psychomotor skills, as it can track users' actions and provide feedback to adjust in real-time. An example found in the literature is on learning how to drive a car using an ITS (Ropelato et al., 2018). The researchers have built a special setup and developed an advanced virtual reality environment for training, as can be seen in Figure 14. The limitation of the virtual reality approach is that it usually requires a special setup (at least the virtual reality glasses, for specific use cases may require extra materials), and the development of such virtual environments may be highly time-consuming.



(a) VR Driving Sim Setup

(b) VR Driving Sim Demonstration

Figure 14. Virtual Reality Driving Simulator using an Intelligent Tutor (Ropelato et al., 2018)

The second category of communication modeling is social intelligence. It refers to the emotional and social connection between tutors and students. Basic strategies in teaching involve understanding and reacting to students' emotions, which can be disengagement, boredom, or frustration. As shown by Taub et al. (Taub et al., 2019), students' affect is directly linked to their learning outcomes while using intelligent tutors. Vygotsky (Vygotsky, 1978) stated that learning

happens in a state of flow, called the Zone of Proximal Development, which is a range somewhere between frustration and boredom, , further described in I.4.3.2. It is highly important for intelligent tutors to be able to model the social intelligence while learning. Some common approaches for recognizing emotions in learning environments (Woolf, 2010) include techniques based on computer vision, physiological parameter analysis and paraverbal language cues.

Visual systems integrate computer vision technologies for enhancing communication modeling in intelligent tutors. Such systems can detect human emotions in facial expressions, hand, and body gestures. In a comparative study between affective and non-affective sensitive tutors (D'Mello et al., 2005), D'Mello et al. integrated non-intrusive sensing devices to detect facial expressions (IBM Blue Eyes camera), a Body Pressure Measurement System (BPMS) for posture information (metabolic indicators) and log files for speech cue recognition from AutoTutor, a constructivist intelligent tutor which can engage in conversations using natural language techniques.

Facial expression recognition is a multi-stage process. The first stage implies the localization and tracking of eye pupils. Then, the system is fitting templates to the upper facial features – eyes and brows. The third step is the labelling of facial action units. IBM Blue Eyes camera technology used by D'Mello achieved close to human expert results, as was shown in their study (D'Mello et al., 2005). Emotions such as anger, disgust, fear, joy, sadness, surprise, or neutral can be detected while learning by using facial expression recognition techniques.

Metabolic indicators used in intelligent tutoring are sensors that can be attached to the environment to recognize emotion. These include, but are not limited to, posture sensing devices, skin conductance sensors, mouses, and cameras. D'Mello's study on affective tutors (D'Mello et al., 2005) presented the Body Pressure Measurement System, which tracks the distribution of pressure on surfaces, such as seats to correlate to students' posture patterns while learning. A specific study on that analyses posture inferring the engagement level of the learner (Mota & Picard, 2003) has presented an experiment that performed real-time classification of nine static postures (sitting upright, leaning back, etc.) with an accuracy of 87.6% on new subjects, and it was correlated with levels of interest (high, low, none) by analyzing posture sequences. Results obtained showed promising future directions on the correlation of metabolic indicators with learning outcomes.

In a study on the psychomotor field, on TUMA ITS (Hodaie et al., 2018) - the tutor for manual procedures activities – authors argue on the advances in tracking the movement of humans with the help of computer vision techniques and wearable sensors. Recent technological progress supports the development of ITSs in psychomotor training (Hodaie et al., 2018). Also, the driving simulator

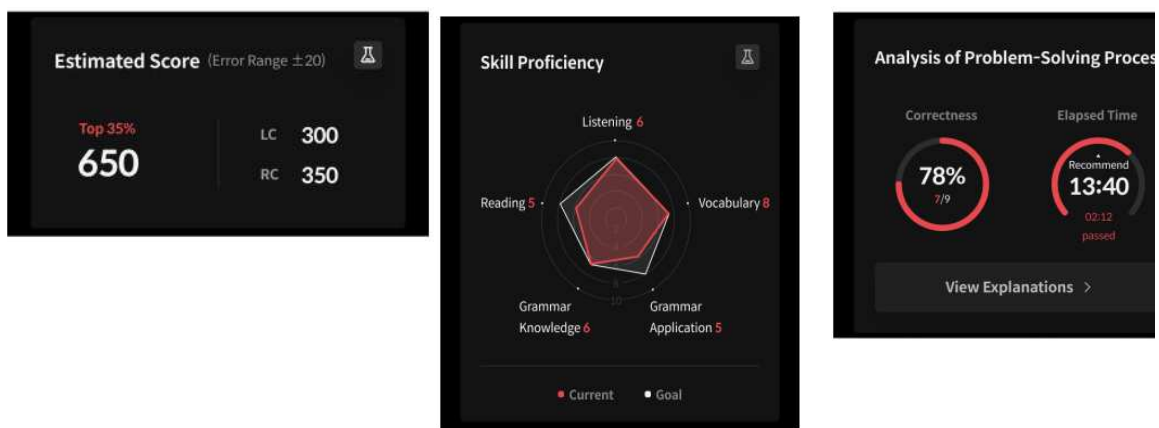
ITS (Ropelato et al., 2018) had a special setup with external devices to track learners' reactions – a driving seat, a steering wheel, and pedals for brake and acceleration.

Speech cue recognition is used to predict emotions in human dialogue using turn-level and contextual linguist features (Woolf, 2010). Through conversation, tutors may depict doubt, confusion, or concentration. The emotions obtained from facial expressions, such as happiness, sadness, or disgust are harder to be understood from speech cue recognition techniques. AutoTutor (D'Mello et al., 2005), through its natural language features, provides rich information about the learner discourse, such as – the correctness of an answer, the verbosity of the student, reaction, and response time, answer length, and other parameters on the quality of response. Using this data, the tutor may provide short feedback, hints, or prompts to advance the conversation.

The third classification of communication modeling is Component Interfaces. It refers to customized user interfaces and user experiences that satisfy the needs of particular intelligent tutors. In a recent work, Kim et al. (Kim, Suh, Heo, & Choi, 2020) challenge the effects COVID-19 pose on education and the success ITS may bring in the educational field. Their work focus on the design of the communication module of an ITS that should promote students' interest in learning, engagement, and motivation through artificial intelligence techniques. The authors have a strong point on the effort of assessing affective interfaces. They underline that usability testing on ITS interfaces done in the literature so far is out of date while the efficiency overall is unreliable since the experiments had didn't test on a large scale.

Kim et al. presented a case study on Santa – a tutoring system based on a multi-platform having over a million South Korean users, available as a mobile app, and website (Kim et al., 2020). Santa is built in the cognitive field, for learning English through listening comprehension and reading comprehension. Santa is mobile-friendly and responsive, and the interface was developed using cutting-edge software. The interface integrates multiple graphical elements to support the learner – score estimation, analysis of the process of solving problems, and proficiency of skills – detailed in Figure 15. Based on the user's initial English tests, which are prerequisites for calibration, the tutor computes an expected performance of the learner, which also ranks him on the overall leader board. Figure 15.a. show an estimated listening comprehension of 300, and a reading comprehension of 3500, which ranks the user in the top 35%. Figure 15.b. shows 2 pentagons; the red has the values of the current user per skill, while the white is the averaged value of all enrolled users. Figure 15.c. displays the likelihood of the user correctly answering a given question and the time required to respond.

Another recent work on learning programming proposes a new approach to communication modeling, through tools and graphic representations – 3D graphics using augmented reality (AR) - which has proven to motivate students and improve comprehension (Schez-Sobrin, Gmez-Portes, Vallejo, Glez-Morcillo, & Redondo, 2020). The authors reason that 2D charts increase the efficiency of learning and have shown historically to increase motivation and student participation in the class. But the 3D graphics show a lot of potential – allowing additional spatial memory capacity through a new dimension and the possibility to understand and easily remember more complex real-world-like elements. They have proposed the use of HoloLens¹¹ and AR techniques for visualizing algorithms and programs and to improve the learning process.



(a)

(b)

(c)

Figure 15. Santa Tutor Interface – "Estimated Score, Analysis of Problem-Solving Process and Skill Proficiency" (Kim et al., 2020)

Authors proposed an innovative approach to learning programming, by creating graphic representations with road signs to facilitate the learning – the process has been named ANGELA ("notAtioN of road siGns to facilitatE the Learning of programming"). The modular programming instructions and statements are represented as road signs – some examples are shown in Figure 16; a - represents the definition of a function, b - represents an if/else statement, c - represents an iterative statement, d - a the return statement of a function, e - an expression evaluation, f - a break statement, and g - a thread. These concepts were linked to solving more complex problems; for example, they designed a 3D component interface with traffic signs for the Bubble Sort algorithm,

¹¹ <https://www.microsoft.com/en-us/hololens>

which is depicted in Figure 17. A white billboard on the left indicated the current algorithm running at a certain point.

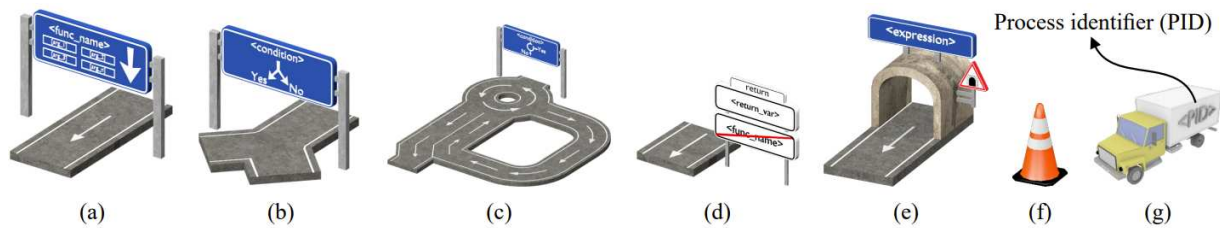


Figure 16. ANGELA Main programming concepts mapped to graphical road signs (Schez-Sobrinio et al., 2020)

The fourth classification of communication modeling is Natural Language Communication. This category is obvious in communication because it is the first interaction model between learners and teachers, it is the most familiar, natural, and does not require any training. Efficient intelligent tutors must handle written and spoken dialogue, they must understand and reason with learners to provide proper teaching.

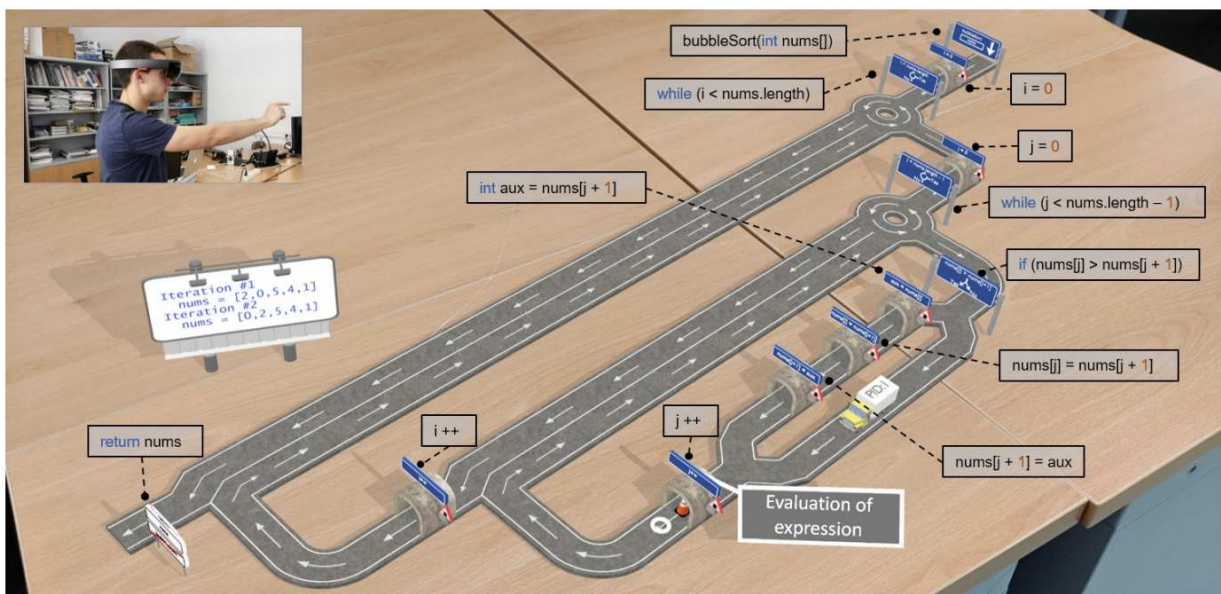


Figure 17. ANGELA "Bubble Sort" algorithm visualization using HoloLens in the 3D Environment (Schez-Sobrinio et al., 2020)

The ultimate goal of a natural language tutor is to create systems that can "think" and "speak" like humans and understand complex communication, including comments, explanations, and discussions (Woolf, 2010). Natural language tutors are classified as follows (Rosé, 2000):

- "Mixed-initiative dialogue" – the conversation is initiated by the student or by the tutor; these are the hardest tutors to build;

- "Single-initiative dialogue" – the tutor considers the former and the latter statements of the student; the initiative belongs to the tutor;
- "Directed dialogue" – the tutor controls the flow and displays relevant information for the students; the tutor can understand brief student responses;
- "Finessed dialogue" – a menu mediates the input yielding a dialogue simulation, and the learner is guided through graphical choices.

A recent bibliographical survey targeting ITSs capable of dialogue based on NLP (Paladines & Ramirez, 2020) shows that most ITSs built in the past 25 years use dialog to help students learn how to solve a problem by applying rules, laws, and others. Most of the natural language tutors were built in the STEM domains – science, technology, engineering, and mathematics, and they use the tutoring modeling based on expectations and misconceptions. The articles found included tutors built for elementary school, high school, and college – from an educational point of view, but also in other environments, which were not reported.

The review found that the most common instructional approach for the NL tutor was the generation of explanations to justify the solutions (54.55% of the tutors), followed by support for problem-solving (30.30%), then clarify and direct procedures (9.09%) and ask questions-answers (6.06%). The way the learner interacts with the system is through text – long answers (69.7%) and short answers (30.30%).

The main communication resources to interact with the learners are animated agents (44.12%), images (38.24%), audio and video (32.35%), simulations (20.59%), or options menus (17.65%). The purpose of the dialogue in the identified natural language tutors is presented in Table 10. As can be seen, most tutors were built to support the explanation of concepts (27.27%), or to justify the solutions (24.24%).

Table 10. Literature review on Natural Language Tutors (1998-2018) – Purpose of the dialogue (Paladines & Ramirez, 2020)

Purpose of Dialogue		Frequency	Percentage (%)
"Understand level"	"Explanation of concepts"	9	27.27
	"Short answer questioning"	2	6.06

	"Explanation of executed actions"	1	3.03
"Apply level"	"Justification of solutions"	8	24.24
	"Application of rules, laws, and theorems"	5	15.15
	"Execute an exercise"	3	9.09
	"Resolution based on schemas"	2	6.06
"Analyze level"		1	3.03
"Create level"		2	6.06

As the literature review has shown, most of the ITSs which integrate natural language communication were built using classical teaching modeling strategies. Also, there was no work reported on non-cognitive tasks, and no study found used apprenticeship modeling or constructivist theories as a teaching strategy.

4.4 Principles of Models Evaluation

Evaluation is the process of collecting and analyzing data to inform decision-making for specific purposes. It involves turning relevant information into meaningful insights that can be used to make informed decisions (Mark & Greer, 1993). Evaluation is different in several fields. For example, classroom teaching evaluation seeks to demonstrate improved learning outcomes, while software evaluation aims to demonstrate that the software works (Woolf, 2010). Intelligent tutoring systems evaluation involves both learning outcome effectiveness, software usability, and evaluation parameters, including learning theory contribution and user experience. The ITS literature lacks a detailed focus on usability testing, usability evaluations, and the user interfaces design process in the development phase (Chughtai, Zhang, & Craig, 2015).

In the early stages of Intelligent Tutoring Systems, researchers were more concerned with the investigation of implementation issues raised while constructing ITS components and modeling the communication between them. Based on that, a comprehensive research base currently exists in

areas such as knowledge representation, teaching strategies, and authoring tools. In the early stage of ITSs, little attention has been paid to evaluation (Mark & Greer, 1993). Recently, more and more ITSs are developed and introduced into educational settings, businesses, or military training environments, and so rigorous evaluation methodologies and techniques are required.

A recent systematic literature review on evaluation methods used in ITSs (Mousavinasab et al., 2018) shows that evaluation mainly involves the measurement of system performance, learner's performance, and experiences. This recent study considered papers published in the most common AIED databases (Web of Science, ProQuest, PubMed, Scopus, Embase, Cochrane, and Google Scholar) between 2007 and 2017 (10th October). The search query defined was based on PICO criteria (population, intervention, comparison, and outcome), which are considered fundamental in evaluation, and includes . The evaluation methods used in the papers found can be seen in Figure 18. Most of the studies were used in computer programming (55%), then health/medical (15.09%), followed by mathematics (15.09%). The focus on the health/medical ITSs is also in the cognitive field, for use-cases in theoretical education such as anatomy, childhood diseases, physiology, or clinical reasoning.

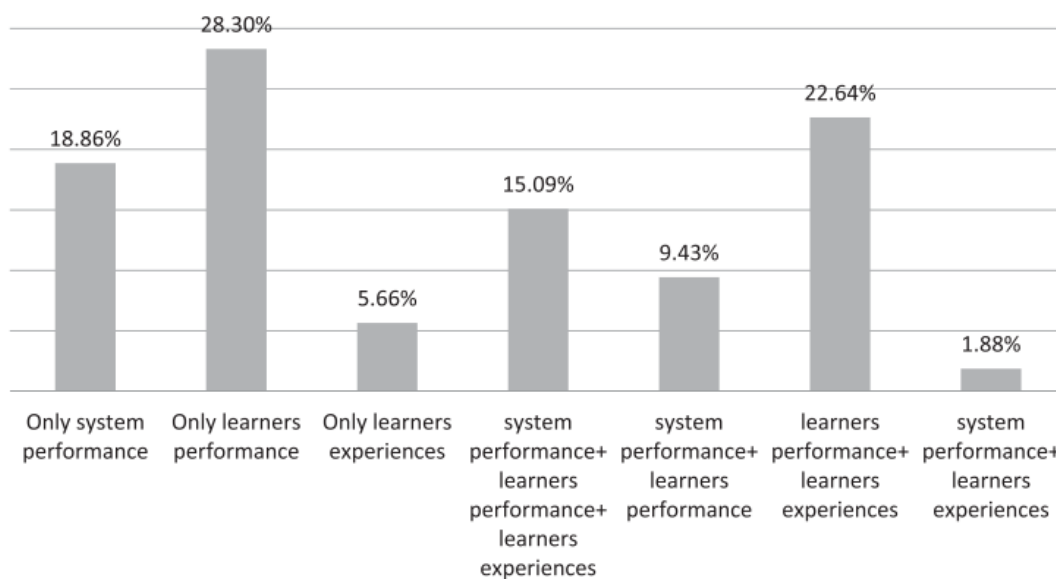


Figure 18. Evaluation methods used in ITSs between 2007 – 2017 (Mousavinasab et al., 2018)

According to a study (Mousavinasab et al., 2018), learner performance refers to the ability to apply procedural knowledge to tasks such as problem-solving, decision-making, and algorithm thinking. As Figure 18 shows, learners' performance was the most common evaluation method used in ITS (28.30%) but also used in joint with learners' experience (22.64%), or in joint with both system performance and learners' experience (15.09%). Evaluating learner experience is a method that focuses on the usability of the system and the overall experience of the user while using it. This

evaluation looks at any issues or problems related to the system's usability. System performance evaluation includes several criteria such as accuracy, precision, adaptability, reliability, sensitivity, or recognition rate. Although mobile devices have the potential to revolutionize the way ITSs are built, the transition from web-based systems to mobile ones has been slow. The literature review shows that the user interface infrastructure was mainly built on web-based interfaces (55.1%), while the mobile interfaces have been implemented in 15.09% of the studies, with fast growth in the past years. All the studies presented target the cognitive field.

Mark et al. proposed a classification of evaluations into formative and summative (Mark & Greer, 1993). Formative evaluation is a type of assessment that occurs during the design and early stages of a project's development. Its primary purpose is to gather feedback and make adjustments to the system while it is still being developed, in order to improve its functionality. This evaluation is often used to obtain information that can help optimize the performance of an ITS. Summative evaluation is performed on the final, stable systems and it is used in making claims about such systems. It aims to address more fundamental questions such as "In what manner does an ITS influence student's education? ", or "Does an ITS has the predicted learning outcome? ".

Evaluation of an ITS should take into account both short-term and long-term considerations. According to Woolf(Woolf, 2010), there are six stages of tutor evaluation that should be followed, including defining the goals of the tutor, establishing the goals of the evaluation, designing the evaluation process, implementing the tutor, analyzing the results of the tutor, and discussing the evaluation findings. In the next section, the evaluation principles will be described further, followed by a few recent ITS evaluation use cases, in both the cognitive and psychomotor fields.

4.4.1 Tutor Evaluation Principles

There is extensive research demonstrating that educational software can be an effective tool for enhancing learning, often surpassing the benefits of traditional classroom instruction or other teaching methods (Woolf, 2010). A gold standard in arguing for the efficiency of intelligent tutors is the success of one-to-one teaching done by human educators. The students who followed the one-to-one tutoring strategy had better results than 98% of the students who learned in traditional classrooms (Woolf, 2010). The potential of intelligent tutors' efficiency is driven twofold. On one side, such tutors generally promote a form of one-to-one tutoring, which has been proven efficient. On the other side, technological advancements provide a good opportunity for enhancing learning outcomes and engaging users in learning activities.

The first phase of evaluating an intelligent tutor is identifying the tutor's objectives. Some ITSs aim to teach specific knowledge (mathematics, physics, computer programming), learn new skills, and

procedures, or use certain equipment (how to drive a car, how to perform training robotic-assisted laparoscopic surgery, how to pass a ball and shoot, or how to use a weapon). The nature of the tutor's goal drives the definition of the expected learning outcomes. Intelligent tutors might measure the transfer of skills between domains, the improved learner self-efficacy, or the updates of a student's attitude about a domain (Woolf, 2010).

The second phase implies identifying the goals of the evaluation. As shown in Figure 18, different tutoring systems have different goals, they can have one goal or more. It can be either learner performance, learner experiences, system performance, or a mix of them. Other evaluation objectives might include evaluating specific components of an intelligent tutor system, such as student, tutor, or communication models. For student modeling, it may make sense to measure the predictive power of the model or if it can track and record student knowledge accurately. In the tutoring modeling, one may measure the tutoring capability to keep learners engaged, challenged, and not overwhelmed. Evaluating communication models can involve assessing the effects of different communication methods, such as virtual reality or natural language techniques, on student learning and motivation.

Studies focusing on tutors' evaluation need to describe the testing environment, and a comprehensive profile of the students involved in the learning experiment – prior knowledge, age, relevant characteristics, and potential confounds to consider. Evaluation processes may be altered by bias and common problems in data collection. In the goals evaluation phase, underlining potential bias sources makes the experiment setup clearer and the common problems are easier to control. The first level of bias can be introduced when the student testing group is created. Students need to be well-balanced and relevant for the evaluation goal – prior knowledge, mean of IQ, income, ethnicity, etc. Then, bias may be introduced in the choice of treatment. If for example two groups of students are involved in an experiment, one of the groups using traditional teaching strategies – called the control group, and the other using intelligent tutoring systems, and one of the groups receives additional materials or equipment, the results will be biased.

The third phase implies the design of an evaluation based on tutor goals and evaluation goals. When building an evaluation methodology, factors such as time – when it takes place, space – what is the experiment location, or how many people are involved should be taken into consideration. The methodology should consider strategies for summative and formative evaluations. Even though the formative evaluations are important, the focus of most tutors is on the summative evaluations, which aim to assess the overall efficiency of the tutors (Woolf, 2010). Formative evaluations are used for catching errors in development or assessing user feedback on the tutor interface, which can be

confusing, or overwhelming. This can be achieved through either questionnaire, interviews, or observation.

There are several types of evaluation designs that can be used in the assessment of intelligent tutor systems. These include pre-test, intervention, post-test, delayed post-test, interrupted time series, crossover, and partial crossover. Pre-test measures assess learners' knowledge level prior to starting the experiment and may also evaluate student characteristics such as learning style or motivation to ensure an even distribution of subjects across groups. The tutor intervention phase involves the tutor actively participating in the learning process with specific teaching goals. The post-test is conducted at the end of the experiment to measure the knowledge acquired by the learners. An example of an evaluation design that includes pre-test and post-test is shown in Figure 19. a. Delayed Post-test is used in evaluation design to measure the long-term effects of learning. The evaluation setup with pre, post, and delayed tests can be visualized in Figure 19.b. The issue with these approaches is that they do not track the moments when learning happens, when learners improve their skills, and what are the roots of learning – see Figure 19.c. This is required to be able to improve the tutor, to assess which features worked, which did not, or which teaching strategies are efficient or not. The next method is addressing this specific issue.

Interrupted Time Series implies the measurement of learning outcome through repeated post-tests, which will enable the assessment of differences in learning – see Figure 19.d. Even though there are high benefits of this method, it is time highly time-consuming, and it involves more work to be enabled. The crossover method, illustrated in Figure 19. e., involves a more complex setup with four groups of students, two forms of intervention (such as intelligent tutors and traditional classroom instruction), and two versions of the same test (test A and test B). The process involves each group first receiving one type of intervention, followed by a pre-test A and then a post-test A. The group's intervention is then switched, and they complete a pre-test B and a post-test B. This method can assess the effects of different teaching methods in the population selected. The disadvantages of this method are the complicated setup and the complexity of the four experimental conditions. The partial crossover method (Figure 19.f) is a simplification of the crossover method, with only two groups, but following the same rules. This method is simpler to put in place, but it will reveal less knowledge on the learning effects of interventions.

The evaluation design can be performed for real-world or laboratory setup. Real-world experiments are preferable to laboratory environments, as they increase the tutor efficiency argumentation. On the other hand, laboratory tests are useful in certain scenarios, as they permit experiment control, such as the number of subjects, prior knowledge, and learning profiles. When designing an evaluation, a critical factor is describing its validity, or what the experiment will measure to consider

the evaluation success. If a hypothesis made during an experiment holds true for a larger population and can be generalized to the outside world, the evaluation has external validity. On the other hand, if there are issues with the tutor design or the evaluation process itself within the same experiment, the evaluation has internal or empirical validity.

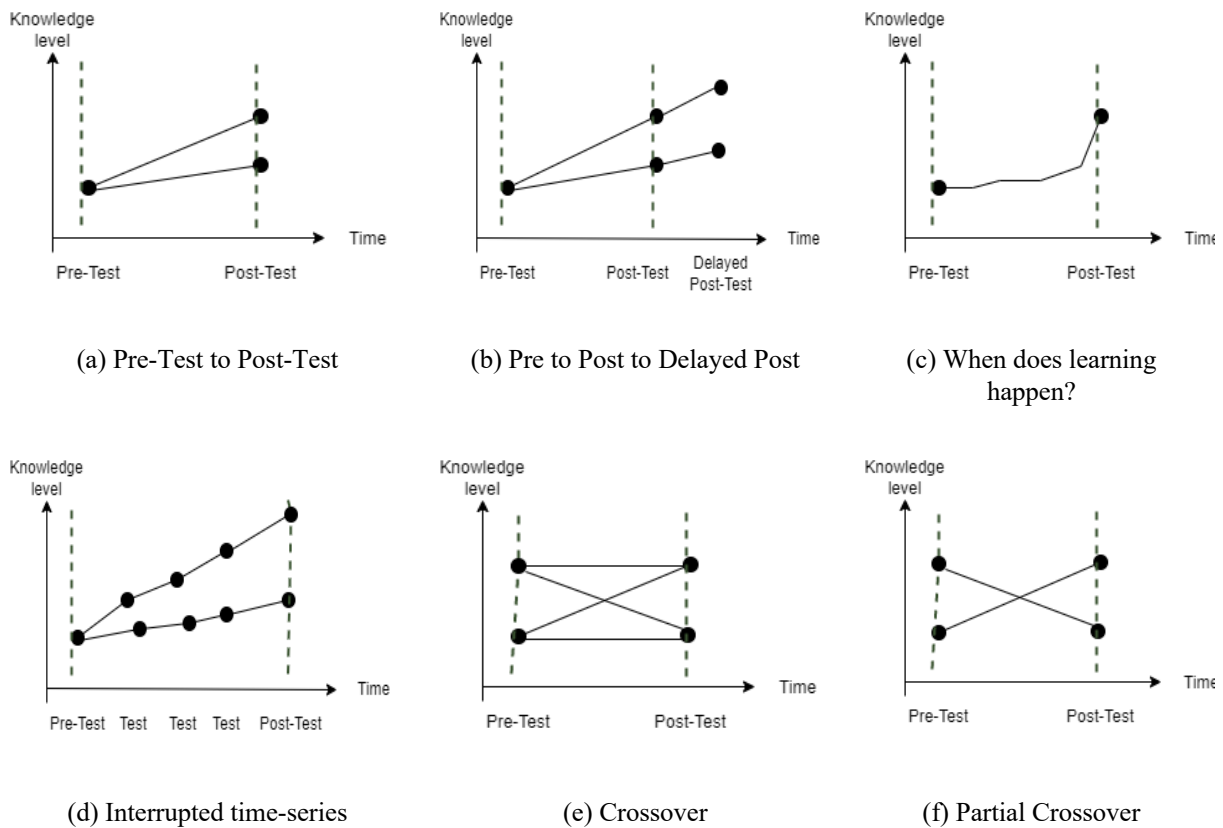


Figure 19. Evaluation designs schemas

Evaluation efficiency needs to be performed using benchmarks. Woolf (Woolf, 2010) proposed six examples of evaluation comparisons, with their associated prototype designs, adapted from the theories of evaluation design, which are summarized in Table 11. The tutor-alone approach has no comparative data, so it is very limited. This approach can measure, for a group of students, only if there is a learning outcome on post-tests. A non-interventional control group is a group of students who receive no teaching. When comparing a tutor with a non-interventional control group the evaluation aim is measuring if an intelligent tutor is better than no tutor at all. One of the most common evaluation comparisons is tutor versus traditional classroom. The goal of this type of evaluation is to determine whether intelligent tutors are more effective at promoting learning than traditional teaching methods. The disadvantage of the discussed approach is that it cannot identify which tutor features lead to learning. This setback is addressed by the Tutor 1 versus Tutor 2 comparison, where groups of students use different versions or teaching strategies of the same

system. Goals of such a method may be to find the best approach to model a specific component of the system – student, communication, or teaching.

The tutor versus ablated tutor comparison aims to identify which are the key tutor feature that contributes to the learning outcomes. For achieving this, the comparison is performed between the initial tutor and a version of the tutor which lacks a specific component or feature. Through this, it is easy to measure the learning benefit of tutor features, such as natural language communication, or eye-tracking mechanisms. The last comparison proposed is Tutor A versus Tutor B, where distinct intelligent tutors are compared. The tutors may encode different techniques for knowledge modeling, encoding, or representation. This method is considered one of the most rigorous ways of intelligent tutors' evaluation.

Table 11. Evaluation Comparisons Examples and Designs Used with Intelligent Tutors (Woolf, 2010)

Evaluation Comparisons	Evaluation Designs
"Tutor alone"	"Intervention + Post-test"
"Tutor versus non-interventional control"	"Pre-test + Intervention + Post-test"
"Benchmark: Tutor versus traditional classroom"	"Pre-test + Intervention + Post-test + Delayed Post-test"
"Within system: Tutor 1 versus Tutor 2"	"Interrupted Time Series"
"Tutor versus ablated Tutor"	"Crossover"
"Between systems: Tutor A versus Tutor B"	"Partial Crossover"

The fourth phase of tutor evaluation involves implementing the evaluation design that was developed in the previous phases. This phase involves fleshing out the details of the experiment, including defining the dependent and independent variables, determining the number and type of participants, providing a detailed description and justification of any control groups, and evaluating the usability of the software. The overall goal of this phase is to provide a detailed description of the experiment based on the decisions made in the earlier phases.

Dependent variables are typically connected to the goals of the tutor, which may include measuring the acquisition of knowledge or evaluating the effectiveness of the tutor (Woolf, 2010). Examples of dependent variables include:

- Learning achievements: pre-tests and post-test to assess performance;
- Learning efficiency: compare if the time spent using the tutoring system is smaller or greater than traditional learning;
- Students' affects: check learners' overall experience and feelings while using the system;
- Feature efficiency: check which features were used and what was the gain.

Other dependent variables are tutor performance, tutor accuracy, skills developed, and skill retention, through delayed post-tests. The evaluation goals should be linked clearly with the dependent variables and measurements should be performed accordingly.

Independent variables are collected through human-system interaction and are commonly used to see if there is any link between them and dependent variables. Independent variables include relevant log data about learners, such as number of completed tasks, time on tasks, common learner errors, number of hints, examples read and time spent on them, and others. Other independent variables are process data, which include discourse information – what the learner said, time spent on dialogue, or dialogue turns, and also nonverbal behavior which includes an affective measure and external sensors for either heart rate, body movements, gestures, or eye movements.

The target population selected for the evaluation study should be relevant, and the group of learners must comply with the evaluation goals. The population is usually split into two categories – the control group and the treatment group. On one side, a control group is defined as the population, which is using the existing, or known teaching methods. On the other side, the treatment group is the population that is testing the novel teaching methods, the new intelligent tutoring system, or the new teaching method or feature proposed.

Learning control measurement should be guided by principles and should be grounded in theoretical perspectives on performance (Woolf, 2010). In some cases, such measurements are not possible, which may drive to studies with non-concluding evaluation results. By principle, evaluation should be performed blind. Both learners and evaluators should not know in which category they are, which category is currently testing, and what is being measured. In addition to tutor efficiency, a parallel evaluation should be performed on tutor usability, which includes the learners' perception of the easiness of use, responsiveness, design, interactivity, or familiarity.

The fifth phase of tutor evaluation involves presenting the findings of the study. This typically includes tables, graphs, and statistical tests. The population used in the experiment includes a set of persons with common characteristics relevant to the study. A sample of the population is a sub-set from the population with identical characteristics. The sampling distribution is the collection of unique samples from a population. The learners' knowledge levels are estimated through different tests, performed before, during, or after tutor intervention.

The last phase of tutor evaluation is evaluation discussion. This phase will outline the anticipated and unanticipated results, problems found, bias, confounding variables, and a comprehensive interpretation of the results. The discussion usually presents the wider implications of the studies, future work, and limitations of the current experimental setup. The next section will present relevant use cases of intelligent tutors' evaluations, which a short description of each implementation phase.

4.4.2 Evaluation Use-Cases

Here, I will present a few tutors' evaluations in both cognitive and psychomotor fields. In many psychomotor tutors, no rigorous evaluation is performed. For example, TUMA ITS (Hodaie et al., 2018) presents the system architecture, but no evaluation is described. For the Virtual Reality Driving Simulator (Skinner et al., 2018) an evaluation is presented, but only on the user experience related to the immersive VR, not on the effectiveness of the system. A more rigorous evaluation was found in the GIFT project-related studies (Goldberg et al., 2018), where different approaches were tested to measure and diagnose novice errors across fundamental principles of rifle training.

The recent literature review on evaluation methods (Mousavinasab et al., 2018) found a list of 53 intelligent tutors published between 2007 and 2017 in the cognitive field, targeting several educational fields, and which followed an evaluation methodology. The selected studies target different populations, mainly school students and university students from different fields of study (computer science, engineering, languages, informatics, medical), but also medical personnel, or pathologist residents. Some studies do outline the population description.

A literature review found tutors which were evaluated to improve learning in disciplines such as mathematics, computer programming, human circulatory, cardiovascular physiology, basic electronics, electricity, Islam religion, chemistry, and molecular biology. Evaluation types include system performance, system adaptability, system usability, student performance (through pre-tests, post-tests, experimental and control groups), and user experiences.

PHP ITS (Weragama & Reye, 2014) was one of the studies targeting computer programming skills. Artificial Intelligence techniques, more precisely Bayesian networks, were used to determine and

update the student model. The learner's characteristics were quantified by his responses to learning activities. The responses are PHP programs written and submitted by students, which are converted by the ITS into a set of facts and then compared to the overall goal to see if it matches or not. Two types of evaluations have been conducted. The first evaluation focused on student performance, using a pre-test and a post-test and dividing the study population into experimental and control groups. A second evaluation has been performed for learners' experiences.

Two groups of postgraduate students at the Queensland University of Technology were involved in the experiment which took two semesters, starting in September 2012. A first version of the system has been tested in the first semester – with 19 students. Then, an improved version, based on experiences, feedback, and discussion has been made available for the second semester, which involved 15 students. Evaluation implied students to work on PHP exercises using the intelligent tutor, with support from external web links. The system recorded learners' actions continuously. Evaluation comparison was within the system – Tutor 1 versus Tutor 2 and it showed that students have learned PHP using the tutor, and the second version was significantly better, as can be seen in Figure 20.

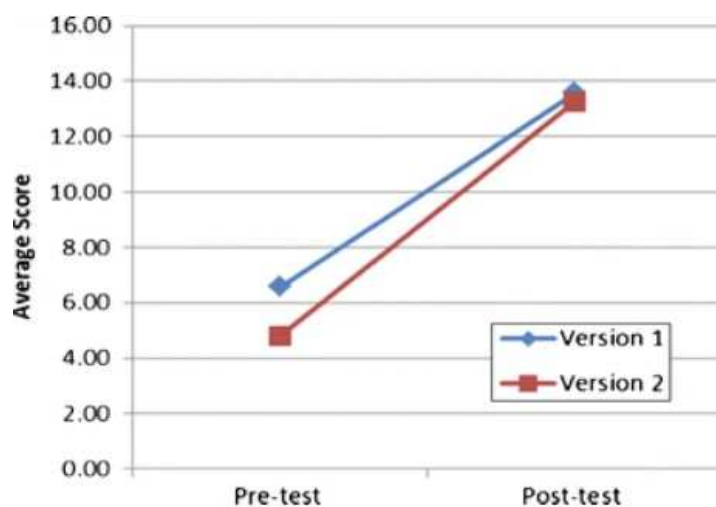


Figure 20. PHP ITS. Tutor 1 vs. Tutor 2 Evaluation (Weragama & Reye, 2014)

Another example of an effective Intelligent Transportation System (ITS) is TECH8 (Dolenc & Aberšek, 2015), which was tested in Slovenia on a curriculum focused on gears in the Technology and Science class in primary school. This system uses intelligent agents to provide adaptive learning content and path, recommendations, and feedback. The study population was composed of school students, and learners' characteristics such as knowledge and learning capacity were tracked. The evaluation only looked at student performance and used experimental and control groups. TECH8 also has a system for collecting and analyzing metadata and variables (SCAMV) which demonstrated that the system was successful in collecting this data.

The results of the experiment are shown in Figure 21 and summarized in Table 12. The high standard deviation in both groups indicates significant differences in the learners' knowledge levels. The negative value of skewness in the experimental group indicates that the majority of the data points are concentrated on the left side of the distribution, while the positive value in the control group indicates that the majority of the data points are concentrated on the right side. These findings suggest that the TECH8 system was effective, as the majority of students in the experimental group had good results, while the majority of students in the control group had poor results.

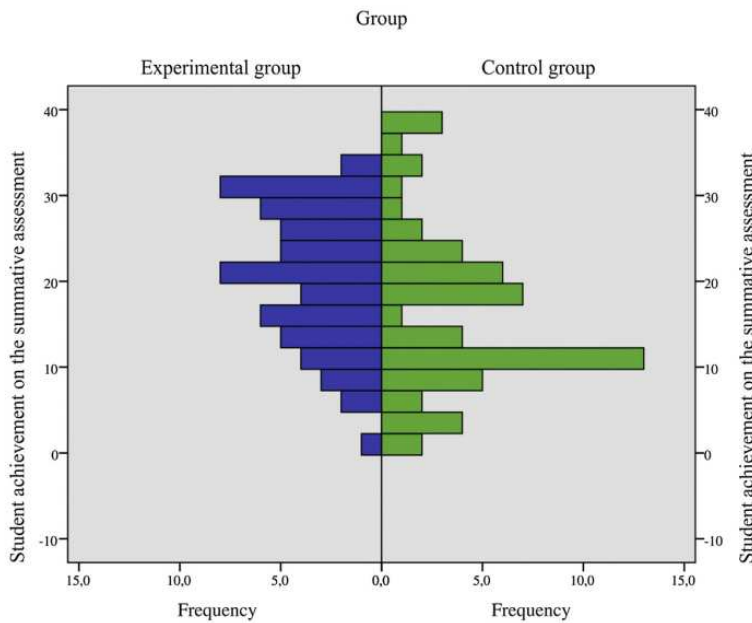


Figure 21. "The frequency of results for the control and the experimental group." (Dolenc & Aberšek, 2015)

Another study in the cognitive field is presenting a multi-armed bandit ITS for mathematics, more specifically decomposing numbers process, aiming to support 7-8-year-old schoolchildren (Clement et al., 2015). The system uses empirical estimates of learning progress based on specific activities provided to students. The optimization process is developed through reinforcement learning techniques, which efficiently manage the exploration and exploitation challenges. The evaluation has been performed in two phases. The first evaluation is performed with systematic simulations, while the second evaluation uses a population of 400 schoolchildren.

Table 12. Learners' evaluation results in the summative assessment. (Dolenc & Aberšek, 2015)

Group	Mean of the total score		Number of students	Mean of total score %	Max Points	Standard deviation in %	Skewness

Experimental	20.71		59	55.70 %	39	20.78 %	-0.236
Control	16.72		58	44.30 %	39	25.11 %	0.657

Simulations were conducted with two categories of population: "P," which had four different distributions representing different levels of understanding for each parameter, and "Q," a population with higher levels of understanding. Three approaches are evaluated – Expert Sequence, ZPDES (based on Zone of Proximal Development), and RiARiT (based on Multi-Armed Bandits implementation). In the long term, RiARiT provides the best learning outcome for both populations, as can be seen in Figure 22. The red and black markers on the curves mean that the differences are statistically significant. The user study performed involved 11 schools in Bordeaux, France. The 400 learners were split into 4 groups. The first one was the control group and the other three were the teaching strategies proposed – the Expert Sequence group, ZPDES group, and RiARiT group. The evaluation type was based on learners’ performance. A pre-test and a post-test were performed with all the groups. The results of the second evaluation indicated a significant increase in the learning speed for several learners' competencies and a high potential for personalization with minimal information.

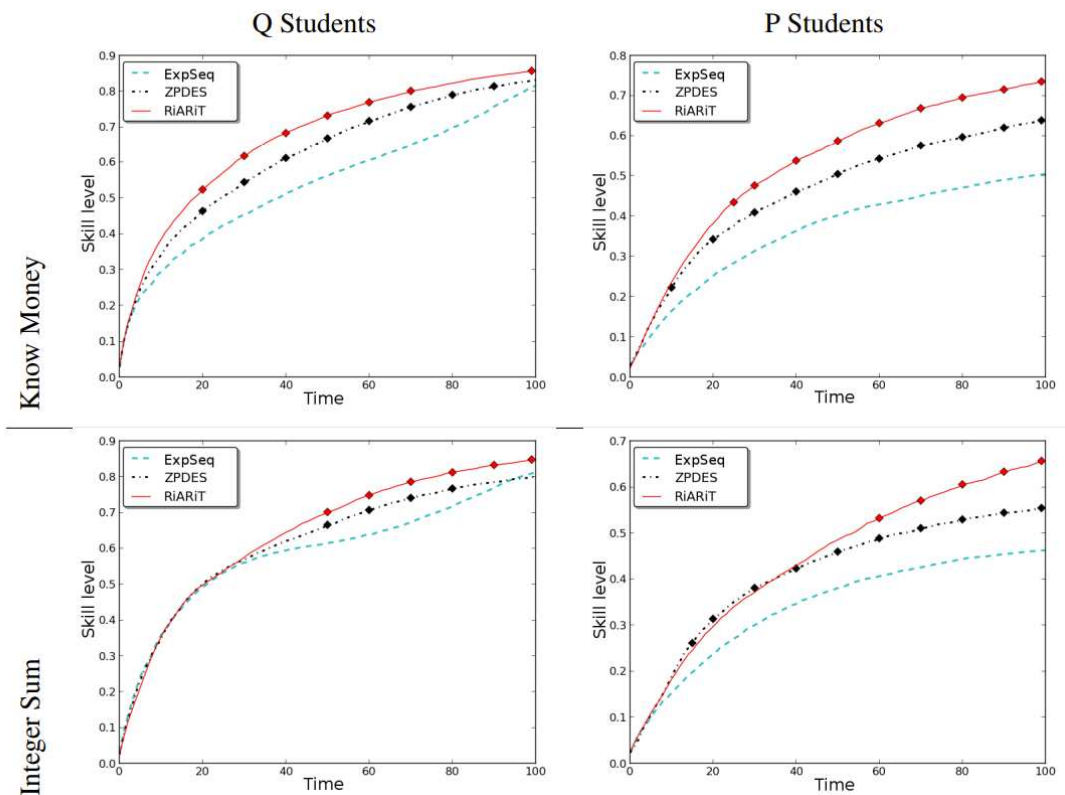


Figure 22. Skill's Levels Evolution on two simulated populations "P" and "Q" using different teaching approaches (Clement et al., 2015)

In the psychomotor field, based on my findings the evaluation for most of the systems built does not follow any evaluation methodology. TUMA ITS (Hodaie et al., 2018) is a study on manual procedural activities, presenting an architectural proposal and outlining solutions to known challenges in the field, but no evaluation method is mentioned. The ball-passing intelligent tutor (Lee & Kim, 2010) was evaluated with an experiment in four episodes. Each episode has an incremental difficulty, from novice to expert. The experiment does not specify the number of participants or population type but based on the figures, it seems like only one person has been testing the system. The authors argue that the user improved the ball-passing capacities using the proposed system.

Training in Robotic Assisted Laparoscopic Surgery (RALS) using an intelligent tutor (Skinner et al., 2018) involves using a simulator to practice robotic surgery procedures. The tutor acts as a controller for the simulation. Research has shown that simulation-based surgical skills development is effective and has the potential to produce similar learning curves to non-simulation learning. Four simulation systems are available for the development of specific skills, based on a standardized curriculum, and three of them are shown effective to improve basic robotic skills. Authors focused on a specific curriculum, developed, and validated by specialists in surgeries, called the Fundamentals of Robotic Surgery (FRS) curriculum. This study proposed a new method for task decomposition, which was validated by Florida Hospital Nicholson Centre (FHNC). The authors did not specify how the system was evaluated, but stated that surgeons were able to complete the learning tasks.

The Virtual Reality Driving Simulator (Ropelato et al., 2018) is built on top of ITS architecture. The tutor assesses the driver's skills in various activities which are continuously tested, activities such as stable driving (on straight roads and curved roads), turning, complete stop, constant speed (without elevation, or up and downhill roads), or driver reaction in traffic. The evaluation was focused on the tutor user experience. The study included 17 participants, 12 males and 5 females, with an average age of 29.5 years and a standard deviation of 8.3 years. All participants had a valid driver's license. They were asked to use the system for 15 minutes and then fill out two questionnaires: one evaluating simulator sickness (before and after the session) and one assessing the feeling of presence. Four participants (23.5%) had to stop due to symptoms of simulator sickness. The simulator sickness questionnaire asked learners to rate their level of disorientation, oculomotor issues, or nausea before and after using the system. The questionnaire about the feeling of presence included 22 questions, rated on a scale from 1 to 7, to assess various user experience indicators (as shown in Figure 23). The median is indicated by the bold black line, and the boxes show the 25th and 75th percentile.

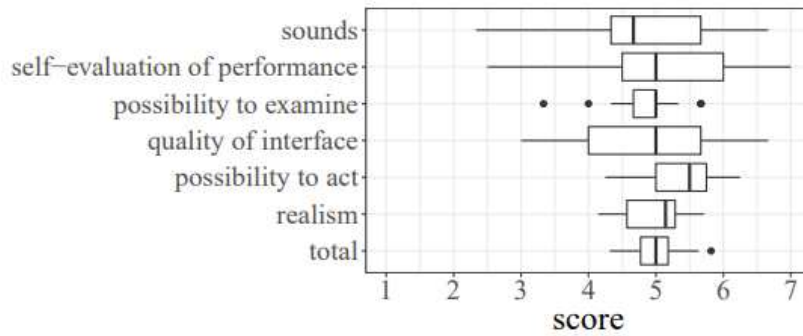


Figure 23. Virtual Reality Driving Simulator ITS User Experience Evaluation (Ropelato et al., 2018)

Overall, the participants reported a satisfactory level of presence, but the study did not measure learning progress or tutor performance in any way. The study concludes that such evaluations are required, and they propose a new direction for future studies on evaluation for the VR driving tutor. The most exhaustive evaluations were performed by the NATO research group, the team which built GIFT, the generalized intelligent framework for tutoring. One study on modeling expert behavior for marksmanship training (Goldberg et al., 2018) aims to measure and diagnose novice errors across fundamental principles of rifle training. They measure behavioral characteristics across each shot – the position of the body, the inspiration/expiration, the pull on the trigger, and attaching of the weapon silencer. The evaluation discovered which behavioral characteristics are linked with the errors done by marksmanship. Even though it is not a direct tutor evaluation, the GIFT research team follows rigorous, methodological evaluations for testing elements linked to GIFT models and proving their efficacy.

The above strategies for knowledge modeling in domain, student, tutoring, and communication modules will be used further in designing our experiments. Based on the theoretical description of each ITS component, we will further propose in Chapter 0 – our domain, student, and tutoring modeling – *OntoStrength*, and in Chapter 9 - our communication modeling integrated in the *Selfit* system. Also, based on the presented evaluation principles and use-cases, we evaluate our system

5 Intelligent Tutoring Systems in Open Environments. Machine Learning in Practice

5.1 Résumé

Ce chapitre décrit les techniques d'apprentissage machine (ML) utilisées dans les systèmes de tutorat intelligents (STI). Ces techniques sont utilisées pour améliorer l'adaptabilité et la flexibilité du système. Les principaux objectifs de l'utilisation de l'apprentissage automatique dans les STI comprennent l'inférence des caractéristiques cachées des étudiants, l'identification des meilleures stratégies d'enseignement, l'optimisation de la réponse du tuteur, la simulation des actions des apprenants et la reconnaissance des comportements imprévus. Une analyse de la littérature sur l'utilisation des ML dans les STI a révélé que la technique la plus courante est la modélisation des caractéristiques des apprenants, suivie par l'orientation adaptative, l'enseignement adaptatif et l'évaluation de l'apprenant.

En outre, un sous-domaine de l'apprentissage automatique est présenté, à savoir l'apprentissage par renforcement (AR). L'apprentissage par renforcement est un domaine d'étude qui se concentre sur la manière dont les agents peuvent apprendre à se comporter dans des environnements afin de maximiser un signal de récompense, et il modélise la manière dont les humains et les animaux apprennent, par le biais d'un processus d'essai et d'erreur. Les éléments essentiels d'un problème d'apprentissage par renforcement comprennent un agent, un environnement, une politique et une fonction de récompense, qui sont détaillés. Les problèmes d'apprentissage par renforcement sont formalisés sous la forme de processus de décision de Markov (PDM) et il existe deux grandes catégories d'algorithmes d'apprentissage par renforcement : en ligne et hors ligne. Les méthodes RL en ligne apprennent la politique tout en interagissant avec l'environnement, tandis que les méthodes RL hors ligne apprennent à partir de données pré-collectées. Le problème du bandit à plusieurs bras est une classe spécifique de problèmes de RL où un agent doit choisir parmi k options pour maximiser la récompense cumulée.

Ce chapitre présente également les techniques de ML utilisées pour le développement psychomoteur. Les réseaux neuronaux artificiels sont efficaces pour calculer la structure des charges d'entraînement, tandis que la modélisation floue avec des algorithmes immunitaires a été proposée pour modéliser l'entraînement sportif. Une récente revue de la littérature sur les méthodes d'analyse intelligente des données pour l'entraînement sportif intelligent a révélé que le domaine de l'entraînement sportif intelligent a gagné en popularité au cours des cinq dernières années et que les études trouvées ont

été regroupées en quatre groupes principaux : Méthodes d'intelligence informatique, exploration de données, apprentissage profond et autres.

La dernière section du présent chapitre présente plusieurs tuteurs intelligents qui utilisent des techniques d'intelligence artificielle pour améliorer l'apprentissage. La plateforme Italk2learn utilise des techniques bayésiennes (classification et raisonnement) pour adapter les messages de rétroaction, améliorer les gains d'apprentissage et détecter les états affectifs de l'apprenant. Un autre système de tutorat, ASTI, a été développé pour améliorer les gains d'apprentissage des étudiants dans les cours d'IA en utilisant des techniques d'exploration de données et des systèmes experts basés sur des règles. Les résultats ont montré des corrélations positives entre les notes du tuteur et les mécanismes de notation automatisés, et 83 % des étudiants ont été évalués dans la bonne catégorie de notes. L'algorithme UCB (Upper-Confidence Bound) pour la sélection des items a été proposé comme une approche plus simple et plus efficace pour l'évaluation formative dans la modélisation du tutorat. L'algorithme UCB a été testé sur 104 étudiants de premier cycle et a montré de meilleures performances par rapport aux méthodes traditionnelles, ce qui a permis de réduire la durée de l'évaluation sans perte de précision. L'algorithme SBTS (Skill-Based Task Selector) proposé par Andersen et al. est un autre système intelligent qui utilise des algorithmes de bandits à bras multiples pour l'attribution personnalisée de tâches d'apprentissage aux étudiants. Un autre tuteur utilisant des techniques de bandits multi-bras pour personnaliser la séquence des activités d'apprentissage a été proposé par Clement et al. Le système a été testé sur 400 écoliers et a montré une amélioration des performances et de l'engagement des élèves par rapport aux méthodes traditionnelles.

5.2 Overview of Teaching Techniques for Intelligent Tutors

Human tutors constantly acquire new knowledge and adapt based on their experiences. This ability to learn is essential for them to advance in their field. The same principle applies to intelligent tutors. If an expert system is not able to learn from its mistakes, it cannot be considered truly intelligent (Selfridge, 1993). The behavior of an intelligent tutor should be updated based on its interactions with learners. Intelligent tutors should be able to learn new information about students, domains, or teaching strategies in order to improve. Machine learning can be used to achieve the goal of developing adaptive tutors.

Machine Learning (ML) is utilized within a system for gathering and integrating knowledge, from several data sources and observations, to improve itself. The improvement implies learning new knowledge, knowledge that is not programmed explicitly (Woolf, 2010). In recent decades, the use of machine learning techniques has significantly increased in instructional systems (Mousavinasab et

al., 2018). The success of ML algorithms relies on the amount and quality of training data (Dlamini & Leung, 2018). In the following section, we review the main usages of such techniques.

5.2.1 Machine Learning Techniques in Intelligent Tutors

In an intelligent tutoring system, the experience can be achieved through storing past tutoring data and using it as a training input for the machine learning model (Dlamini & Leung, 2018). Learning over time is called incremental machine learning and this can be achieved through several techniques, which will be briefly described further. The main benefits of adaptive tutors are the increase in tutor flexibility, reduced cost of building the tutor, and adaptation to new student populations (Woolf, 2010). The inflexible tutors have pre-defined rules, and, during student interactions, they will respond with heuristics defined previously, when the designers and developers did not have complete knowledge about the field. To adapt and improve, such tutors need to be updated continuously, which implies more time and implicitly a higher maintenance cost. Also, in a non-adaptive tutoring system, when new student populations start to learn, refined rules and probably new teaching strategies are required, so an extra effort will be implied. This sets a limitation on the effectiveness of such tutors.

Machine Learning techniques in intelligent tutors are used to reason "outside the scope" of the initial variables that defined the student and tutor models, they provide the opportunity to automatically construct or enhance such models. They have the potential to make significant contributions to the field of educational sciences. Through Machine Learning a system can learn about how students learn, reason under uncertainty, and improve the generalizability of tutoring strategies.

Usage of machine learning algorithms has been found in all the ITS components: domain, student, tutoring, and communication modules. Common goals for using machine learning techniques in intelligent tutors include (Woolf, 2010):

- To deduce invisible student attributes – including confidence and knowledge of a student;
- To identify best teaching strategies – includes strategies to assess what works and what does not, for each learner (hints, written explanations, natural conversation, etc.);
- To optimize tutor response – includes strategies for optimizing pedagogical intervention (how and when to respond to learners);
- To simulate learners' actions – includes the generation of "pseudo students" to model human learning and detect potential tutor problems, before the official deployment to real users;
- To recognize unanticipated behavior – includes the development of incorrect student knowledge models to assess unanticipated behavior.

A recent literature review on intelligent tutoring systems which have implemented machine learning techniques for different goals has found 53 relevant studies between 2007 and 2017, all targeting cognitive development (Mousavinasab et al., 2018). The review included studies in education, training, or educational assistance tools, which have demonstrated the usage of the ITS architecture in their systems. The major educational field found in the review was computer programming, with 55% frequency, followed by medical and mathematics fields.

The review found several goals for using Machine Learning techniques in the system proposed, from adaptive tutor modeling (adaptive feedback, hints, learning path), adaptive student modeling (definition, classification of learner’s characteristics), and adaptive domain modeling. The detailed list of machine learning techniques purposes with their associated frequency found in the review can be seen in Table 13. Modeling learner characteristics is the most common technique (56.60%), and it can be seen the count of all techniques is greater than 100% because some intelligent tutors used more than one adaptive strategy. The learners’ characteristics used in delivering adaptive learning include the following: knowledge level (62.26%), learning performance (52.83%), behavior in learning path (41.50%), learning preference (15.09%), learning style (9.43%), or emotional factors (3.77%). Many ITSs have utilized multiple factors related to students' characteristics to build student models, while some others have primarily relied on only one aspect.

Table 13. The purpose of applying Machine Learning Techniques in ITSs between 2007-2017 (Mousavinasab et al., 2018)

Machine Learning Techniques	Frequency (%)
"Adaptive feedback, hint, or recommendation generation"	56.60 %
"Defining, classification, or updating the learner’s characteristics"	52.83 %
"Learner’s evaluation"	45.28 %
"Presenting adaptive learning material or content"	41.50 %
"Adaptive learning path navigation"	28.30 %
"Presenting adaptive tests and exercises"	5.66 %

The main categories of adaptive techniques with their associated sub-categories are the following (Mousavinasab et al., 2018):

- "Adaptive guidance: Adaptive feedback generation, Adaptive hint generation, Adaptive recommendation generation";
- "Adaptive instruction: Presenting adaptive learning material, Adaptive learning path navigation, Presenting adaptive tests and exercises";
- "Learner's evaluation: Knowledge evaluation, Performance evaluation, Skill evaluation";
- "Define and update the learner's model based on: Learning style, Knowledge level";
- "Classification / Clustering the learners based on: Affect, IQ, Learning style, Learning needs, Characteristics";
- "Others: Communication, Calculation of the level of difficulty of exercises, Classification of learning materials".

Computer programming ITSs found in the literature review used several machine learning algorithms to adapt tutoring: Fuzzy-based techniques (20%), condition-action rule-based reasoning (20%), case-based reasoning (13.33%), intelligent multi-agent (13.33%), and data mining (13.33%). Health ITSs found in the review used the following machine learning algorithms: Bayesian-based techniques (50%), methods based on intelligent multi-agent and NLP. In mathematics, most ITSs implemented decision-making based on rules.

The research indicates that for fields with more structured disciplines, such as computer programming and mathematics, condition-action rule-based reasoning is the most common technique and is well-suited for decision-making and problem-solving. In the medical area, where critical thinking is particularly important, tutors should be designed using rule-based reasoning, but in practice, this approach has not been implemented. For fields that require finding solutions to problems by means of similitude with other known problems, case-based reasoning is the preferred approach.

Machine learning techniques can be classified based on whether a supervisor is present to ensure that the correct behavior is being learned. There are two categories: supervised and unsupervised learning. Supervised learning involves the use of a computational supervisor to monitor the learning process (Woolf, 2010). It is suitable for domains where instances of the expected behavior are provided, and learners can easily resolve the tasks. This implies that both input and output can be observed. Methods from supervised learning used in intelligent tutoring systems include neural networks (for pattern classification), decision trees (for rule-based fields), and Bayesian belief networks (for describing variable relationships through statistics). Supervised learning is fast and

effective at handling noisy data, but it requires large, high-quality training datasets, which can be time-consuming and expensive to obtain (Woolf, 2010).

Unsupervised learning involves studying how systems can learn to represent specific input patterns in a way that reflects the statistical structure of the entire collection of input patterns (Dayan, 2009). In intelligent tutoring systems, unsupervised methods do not provide the correct answer to the tutor during the learning process. Classical methods used in intelligent tutoring systems include clustering (using information retrieval methods) and reinforcement learning. Feedback or rewards are provided at the end of the learning process, but without indicating which actions are correct or which rewards are associated with which actions. Only the pairs input-output with a success function are known. The advantage of such methods is that they do not require large datasets, they learn by exploration. For an intelligent tutor, unsupervised learning methods may use large existing data spaces, such as the data gathered from previous students, to measure success (Woolf, 2010). A disadvantage of such methods is that they require extensive exploration which is expensive in both time and space. In some cases, unsupervised approaches may not learn what was initially intended.

As it was shown, there are several intelligent tutor components with machine learning techniques and the adaptive algorithms are widespread across tutors' models. In the discussed context, the center of interest of the latest studies is on tutoring model adaptation, to present adaptive learning material or content – as per Table 13. 41.5% of the systems built between 2007 and 2017 used machine learning techniques to enable this feature.

Recent progress in adaptive tutoring has focused on identifying the activities that will provide the best learning experience for each individual learner, based on an assessment of their competence levels and progress, with limited knowledge about cognitive and student models (Clement et al., 2015). This approach is based on the following principles:

- Less dependence on cognitive and student models. This principle states that the system must adjust and approximate the individual traits of every learner online. Even when using automatic methods, it can be difficult to identify the relevant parameters that best describe each learner. In this context, the aim is to depart from being depended on already defined typologies;
- Efficient optimization methods. This principle states that the system should not make any specific assumption about how a student learns and rely on exploration and exploitation of different activities to estimate the progress;
- More motivating experience. This principle states that overall, the activities that lead to the most learning progress should be chosen, as they have the potential to provide a more

motivating experience based on the Zone of Proximal Development theory (Vygotsky, 1978).

The creation of such tutors involves the use of multi-armed bandit algorithms in tutor modeling (Clement et al., 2015). Multi-armed bandit algorithms deal with the problem of allocating a fixed, limited set of resources among choices in a way that maximizes expected gain. The properties of the choices are only partially known at the time they are chosen, and they become better understood as time goes on. The multi-armed bandit algorithm is a classic reinforcement learning problem that illustrates the exploration-exploitation tradeoff problem. Common techniques for building intelligent tutors using reinforcement learning algorithms are then outlined.

5.2.2 Building Reinforcement Learners

Reinforcement learning (also known as RL) has great potential for modeling tutoring because it resembles the manner in which humans and animals acquire knowledge by strategies that involve learning through positive and negative feedback based on personal experience. Reinforcement learning problems involve learning what to do, which means how to map a situation to the right actions, to maximize a numerical reward signal (Sutton & Barto, 2018). Sutton defined the RL problems as having the following three characteristics:

- Are closed-loop – learning system's actions influence the later inputs;
- Do not have direct instructions, must discover which actions yield the most reward;
- The actions taken at a certain time affect not only the immediate reward but also all the subsequent rewards.

Reinforcement learning is a highly effective machine learning approach for decision-making in interactive environments. RL algorithms are targeted to identify optimal strategy that dictate the selection of an action taken by agents within a specific context in order to obtain the maximum possible cumulative reward (Ausin, Azizsoltani, Barnes, & Chi, 2019).

Four key elements define a reinforcement learning problem: the agent, the environment, the policy, and the reward function. The environment is the external system in which the agent, or learner, exists, takes actions, and moves from one state to another. The agent can demonstrate long-term successful behavior through rewards, which are provided at the end of each action taken. Typically, the reward is a scalar value that the agent aims to maximize and may represent the degree to which a particular action or state is desirable. The reward function defines the goal of the RL problem (Woolf, 2010) and maps each state-action pair in the environment to a number (positive or negative) called a reward, indicating the desirability of that pair. The policy is the agent's behavior at a given

time and may be a simple function or lookup table that checks previous states and actions taken in those states.

The formal framework of RL agent-environment interaction can be seen in Figure 24. The RL problem is described with the help of a formalism represented by a discrete time stochastic control process, in which the interaction between the agent and the environment takes place (François-Lavet, Henderson, Islam, Bellemare, & Pineau, 2018). The agent begins in an initial state within the environment, $s_0 \in S$, where S is the state space, by obtaining an initial observation $\omega_0 \in \Omega$, where Ω is the observation space. At each time t , the agent takes an action $a_t \in A$, where A is the action space. This action has three outcomes: (i) the agent receives a reward $r_t \in R$ (possible rewards described as a continuous set), (ii) the state moves to the new state $s_{t+1} \in S$, and (iii) the agent acquires an observation $\omega_{t+1} \in \Omega$.

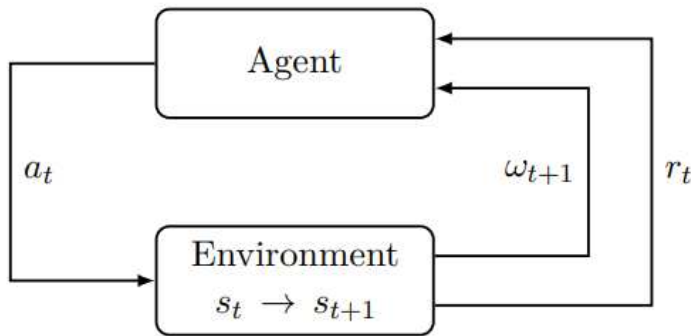


Figure 24. Reinforcement Learning Framework. Modeling Agent-Environment Interaction (François-Lavet et al., 2018)

RL is defined as a Markovian decision process. Markov property is stating that a process depends only on the current observation to foresee the future, not the full history (François-Lavet et al., 2018). Markov Decision Process (also known as MDP) is a "5-tuple (S, A, T, R, γ) , where S is the state space, A is the action space, $T: S \times A \times S \rightarrow [0, 1]$ is the transition function (set of conditional transition probabilities between states), $R: S \times A \times S \rightarrow R$ is the reward function (where R is a continuous set of possible rewards in a range $R_{\max} \in R^+$), and $\gamma \in [0, 1)$ is the discount factor". The illustration of the MDP, used in reinforcement learning, can be seen in Figure 25.

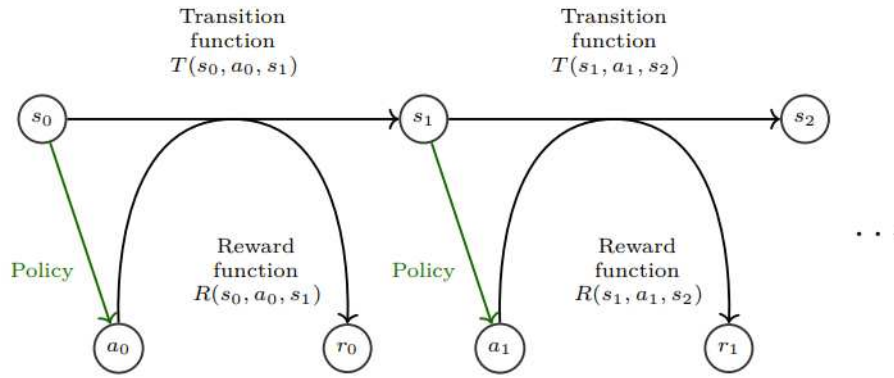


Figure 25. Reinforcement Learning Framework. Markov Decision Process. (François-Lavet et al., 2018)

There are two major categories of RL algorithms – online and offline. Online RL methods learn the policy while the agent interacts with the environment. In the offline approach, the policy is learned from a pre-collected training dataset (Ausin et al., 2019). Online RL is more suitable for domains where the state representation is clear and the interaction with the simulated environment and real environment is computationally cheap and feasible. Offline RL is required for more complex domains, such as e-learning, where human learning problem is complex, and the process is not fully understood.

The multi-armed bandit problem, or k-armed bandit problem, is a specific class of problems in the reinforcement learning field that involves learning how to act in a single situation. It arises when an agent has to frequently select from a set of k distinct actions (Sutton & Barto, 2018). Afterward, a reward represented by a number is provided based on a stationary probability distribution that is dependent on the chosen option. The goal is to maximize the cumulative reward over a period of time.

If the values for each action are known, the problem is straightforward - you would always choose the action with the highest reward. At a certain step, you will know the rewards for a set of actions that were previously selected. One possible future choice would be to choose the one with the highest value, from the known actions. This is called *greedy* action and choosing an action you already know is called *exploiting* the current knowledge. Exploitation, or choosing the option with the highest known reward, is the best choice for maximizing reward in the short term, but exploration of new, unknown options may lead to a higher total reward over the long term.

In any case, the choice to either explore or exploit is dependent in a nontrivial manner on the exact uncertainty values, current estimates, and the number of steps to be covered (Sutton & Barto, 2018). Balancing exploration and exploitation is challenging, and several methods have been proposed. The most common approaches are ϵ -Greedy and Upper-Confidence-Bound (UCB). ϵ -Greedy is a greedy approach of the multi-armed bandit, but with an ϵ - factor of exploration, which can have values

between 0 and 1. If ϵ is set to 0.1, this means that 10% of the time the bandit will explore new actions and 90% of the time will exploit existing actions with the highest reward.

The ϵ -Greedy approach requires non-greedy actions to be tried within a certain ϵ factor but does so without discrimination nor preference for actions that are almost greedy or specifically uncertain. In this situation, the Upper-Confidence-Bound approach may be more efficient. It is recommended to choose the non-greedy action based on its likeliness to become part of the optimal solution, considering the uncertainty and closeness of the estimates to being maximal (Sutton & Barto, 2018). In simpler terms, the options having a lower value estimate or those which have been chosen many times will be less selected by the bandit. The more uncertain the bandit is about a specific arm, the bigger chances it will be selected. UCB is not suitable for nonstationary problems, or problems that have a large state space.

A recent literature review (Mui, Lin, & Dewan, 2021) focuses on the research on bandit algorithms used to support adaptive learning in various settings. Adaptive learning is defined as a set of problems, including knowledge component sequencing, question sequencing, or pedagogical strategies. The authors describe MAB algorithms as data-driven algorithms that can find an optimum point between exploration and exploitation and decide in a sequential manner in the face of uncertain outcomes.

The review identified relevant papers published between 2009 and 2021 and found that multiple problems were addressed using bandit algorithms. Additionally, adaptive learning complexity requires more advanced bandit implementation, based on extended algorithms adjusted to adaptive learning in the online environment. The results, including the problems addressed, the types of algorithms used, and the number of papers found, can be seen in Table 14.

Intelligent tutors have used RL for partially automated construction of student models and to calculate optimal teaching policies (Woolf, 2010). This may include reducing the number of mistakes made while learning or reducing the time spent per each problem. RL methods are efficient in intelligent tutors for cases where testing several policies and populations are required, and this can be achieved by simulating a large number of interactions and students. In such cases, policy function can be improved through simulation, instead of real experience. This is useful for scenarios where real testing is highly time-consuming, or too expensive.

Table 14. Number of papers found based on the bandit algorithms used and the and different challenges in adaptive learning (Mui et al., 2021)

Algorithms	Problems
------------	----------

	Assessment question sequencing	Question Exercise sequencing	Action Task sequencing	KC sequencing
"UCB"	1	-	1	-
"Adversarial bandits"	-	5	-	-
"Contextual bandits"	-	-	5	1
"Gittins"	-	1	-	-
"Epsilon- greedy"	-	-	1	-
"SoftMax"	-	1	-	-
"Stochastic bandits"	-	1	-	-
"Recovering bandits"	-	1	-	-

In a classic intelligent tutoring system (ITS) architecture, the tutoring module uses an estimate of the learner's competence levels and progress to select activities that provide the best learning experience at a given time. The challenge for the tutor is to identify the optimal sequence of activities for maximizing the average level of competence across all targeted abilities (Clement et al., 2015). This challenge is driven by three main factors, which were tackled by Clement et al. (Clement et al., 2015):

- Limited time for practicing activities – the tutor cannot test all combinations of sequences, nor all activities;
- Managing motivation is hard – student engagement in activities is a necessary condition for the effectiveness of learning;
- Individual differences between trainees make an optimal sequence for a given trainee inefficient for another one.

Clement et al. (Clement et al., 2015) proposed a multi-armed bandits approach used in conjunction with the Zone of Proximal Development. ZPD has been defined by an expert, knowledge components are in the mathematics field, more precisely numbers decomposition. The ZPD will be adjusted based on optimization algorithms, taking into account the student's responses and learning progress. The results obtained using multi-armed bandit algorithms are comparable and even surpass, in certain conditions, the sequences created by expert teachers.

5.3 Machine Learning Techniques in Psychomotor Development

The literature and methods related to adaptive ITSs are vast and, in this section, the focus is on the implementations related to non-cognitive tasks, namely, psychomotor skills.

Training individualization is the main condition for its optimization. Coaches often use classical methods for individualization in sports training, such as the "model of the master," which involves calculating the training load based on the volume and intensity of training, or actual training, which involves computing the mean values of training means for a group of athletes over a given cycle (Rygula, 2005). These methods are commonly described in sports training literature. The weak points of these approaches are the lack of individualization, by using tables of standards, and the impossibility to generate new training content, from the existing one.

An approach that uses artificial neural networks to compute the structure of training loads has been proven efficient in a two-year experiment in Silesia (Rygula, 2005), where a group of girls was studied while practicing a 100-meter run. To calculate personalized training loads, coaches often track a set of parameters such as body characteristics (such as height, mass, length, and circumference of the thigh and shank, body composition, and body proportions based on the Rohrer index¹²), dynamic strength (mechanical power) of the lower limbs, anaerobic efficiency, or oxygen efficiency. The training model includes a model and monitoring component to ensure that training is properly controlled. The experiment showed the efficiency of neural networks in computing the proper training load in speed capabilities, demonstrating also, through the parameters used, a direct link between exercises with maximum intensity and the external resistance.

A study using artificial intelligence for modeling sports training in swimming has been conducted (Me & Unold, 2011). This study uses fuzzy modeling based on an immune algorithm to define five levels of training planning: sports career, macrocycle, mesocycle, microcycle, and training unit. The last four of these levels are the ones used by coaches in training planning, and the experiment focuses

¹² https://link.springer.com/chapter/10.1007/978-3-211-89836-9_1249

on the most detailed two training divisions of measurement: microcycle and training unit. The study achieved 70% accuracy.

For the above-mentioned studies, the modeling is done by using sensible data about the trainees (such as body mass, body composition details, heart rate, maximum speed, or anaerobic capacity), which makes it hard to be feasible in an open environment.

In a recent literature survey on intelligent techniques for analyzing data in the field of sports training (Rajšp & Fister, 2020), the authors discuss how current technologies are transforming the manner in which athletes enhance their abilities toward success in next-level competitions. They view sports training as a pedagogical process in which the trainer plays the role of the instructor, planning the trainee's program. Each exercise is described as tasks that involve physical effort and is intended to improve the athlete's sports results. The authors divide the sports planning process into four phases: planning (prescription of appropriate exercise units), realization (execution phase), control (comparison between the athlete's actual exercises and the planned exercises), and evaluation (assessment of the athlete's performance). The phases are interconnected and have a continuous transition, as can be seen in Figure 26.

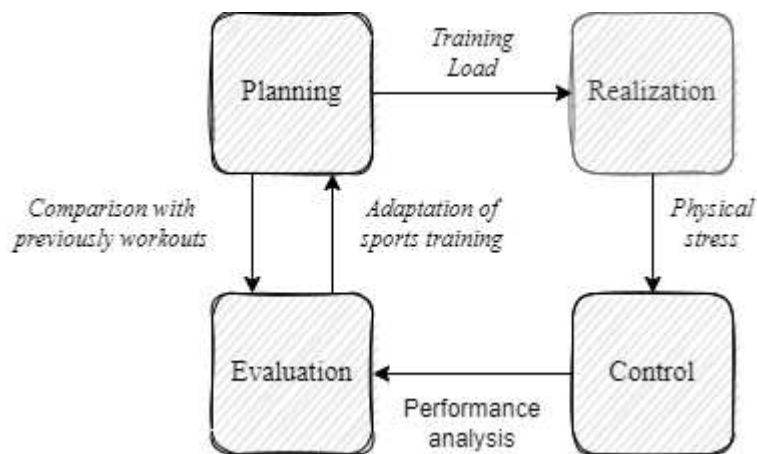


Figure 26. The four phases of sports planning.

The literature review shows that the field of sports training is becoming a popular smart educational field ever since 2016, where the studies found were grouped into a taxonomy split into four main groups:

- Computational intelligence methods (such as simulated annealing, fuzzy systems, evolutionary algorithms, swarm intelligent algorithms like the bat algorithm and particle swarm optimization);
- Data mining (including traditional techniques such as apriori and ML-based methods like support vector machines, decision trees and random forests, adaptive and gradient boosting,

k-nearest neighbors, and artificial neural networks, as well as hierarchical and k-means clustering);

- Deep learning (including convolutional neural networks, long short-term memory and recurrent neural networks);
- Other techniques (such as dynamic time warping, case-based reasoning, Markov chains, Bayesian networks, naïve Bayes, generalized additive models, Gaussian processes, spline interpolation, linear discriminant analysis, and linear and regularized logistic regression).

The study involved a wide range of sports, from soccer, shooting, rowing, volleyball, table tennis, weightlifting, fitness, and others. They were grouped into individual sports (53 studies), mixed sports (17 studies), and team sports (27 studies).

For fitness, the authors found several studies, considering the gym as a great environment for conducting such research. Athletes can easily use various wearables in the form of sensors and devices to accurately assess their exercise data during training - Zhou, Sundholm, Cheng, Cruz, and Lukowicz (2016), Baumbach, Bhatt, Ahmed, and Dengel (2018), Das, Busetty, Bharti, and Hegde (2017).

Many studies present results on classifying the performed motion and counting the number of repetitions of each exercise (Zhou et al., 2016) (Baumbach et al., 2018), as can be seen in Table 15 (Rajšp & Fister, 2020). One study (Fister, Rauter, Yang, Ljubić, & Fister, 2015) is focused on training planification for running. Trainees are using smartwatches to track their distance and heart rate while training. Fister et al. applied the bat algorithm, which is based on variations in the athlete's heart rate, to the planning of athletic training sessions and obtained promising results. Studies focus on one or more research topics, in at least one of the training phases (planning, realization, control, or evaluation). Research topics include injury prevention, prevention, and recovery, training analysis, or classification of the movements conducted.

The studies found do not specify the integration within an intelligent tutoring system, but are either components or systems which support psychomotor development. They are used in different phases of the training program and the methods described can be considered and integrated further into our work.

Table 15. Studies with Smart Sport Tools in the fitness domain (Rajšp & Fister, 2020)

Study	Research Method	Focus	Results	Trainin g	Training Realizatio n	Trainin g	Training Evaluati on

				Plannin g		Contro l	
(Baumbach et al., 2018)	"K-Nearest Neighbors; Support Vector Machine; Bayesian Networks; Decision Trees"	"Comparison of different algorithms used to determine the activity being conducted in the Fitness Center by using data from a smart wrist wearable device. "	"Deep Exercise Recognizer model constructed from received measurements and combined individual Machine Learning models. "	0	2	0	0
(Das et al., 2017)	"K-Nearest Neighbors; Support Vector Machine; Decision Trees"	"An automatic indoor exercise recognition model for both in the gym and home usage scenarios. Classified activities are Biceps curl, Chest fly, Row, Push up, Sit up, Squat and Triceps curl. "	"Accuracy of 95.3% and 99.4% was achieved for activity recognition and repetition count, respectively. "	0	2	0	0
(Fister et al., 2015)	"Bat Algorithm"	"Planning fitness training sessions. "	"A fitness training session generation method that	2	0	0	0

			takes into account muscle groups, intensity, and repetition, so that the balance between muscle groups is achieved for best results in training for a triathlon. "				
(Zhou et al., 2016)	"Adaptive Boosting"	"The demonstration of a wearable system, based on a fabric force mapping sensor matrix, which can measure the muscle movement during various sporting activities demonstrated with the case of leg workout exercises. "	"81.7% accuracy was achieved after 24 different leg workout sessions. "	0	2	0	0

5.4 Examples of Machine Learning-Based Tutors in Other Domains

Many tutors have been built recently, using one or more AI techniques (such as Bayesian network classifying and reasoning, data mining, NLP, Fuzzy-based, and reinforcement learning) applied to several ITS models. Some of the most relevant intelligent tutors will be presented further, based on

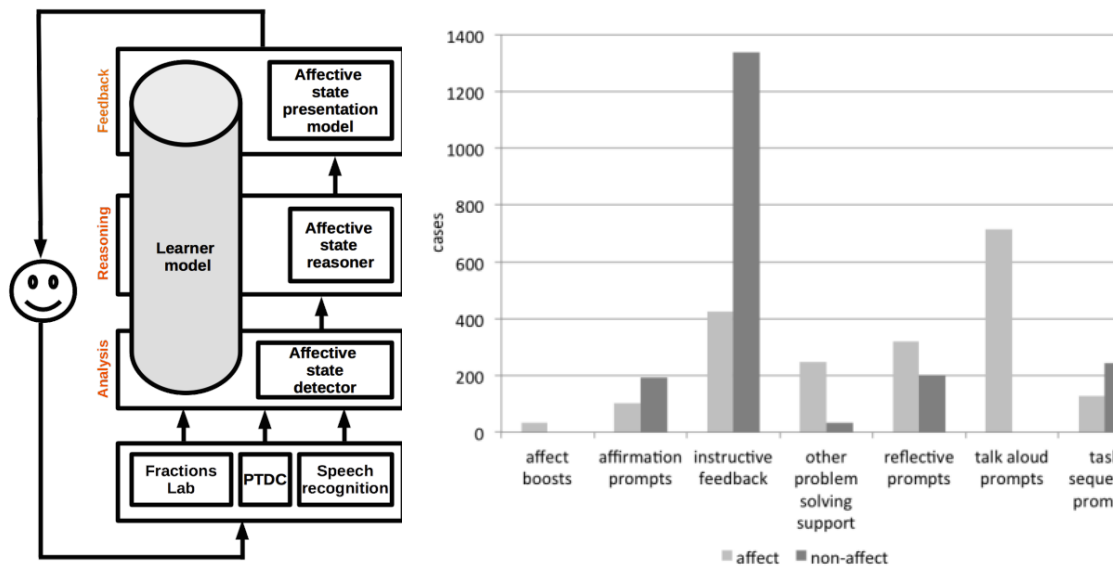
Mousavinasab's study findings (Mousavinasab et al., 2018), but also others found in the recent intelligent tutors' literature.

5.4.1 Unsupervised Learning Cognitive Tutors

A tutor for learning mathematics has been built by Grawemeyer et al. (Grawemeyer et al., 2016) making it part of the learning platform Italk2learn. The goal was to improve learning gains through Bayesian techniques (classification and reasoning) for adapting feedback messages, and feedback presentation, by detecting and measuring affective states and task behavior. The affective state is measured by taking into account the interaction of the learner with the tutoring system.

The data collected includes whether a user has viewed the feedback received, or if the user has followed the advice shown. A classifier for perceived task difficulty, called PTDC, is used to extract features from a student's speech. PTDC's goal is to determine if the learner is over-, appropriately, or under-challenged. A speech recognition software has been used by the authors to produce an array of spoken words, together with the time elapsed between words. A Bayes classifier was applied for classifying the affective states based on the output of the speech recognition software. The adaptive support proposed by this study was made up of three phases: analysis, reasoning, and feedback, which can be seen in Figure 27 a. Authors proposed the following feedback types:

- Affect boosts – enhancing student motivation in solving particular tasks. This includes prompts that were shown to the users to acknowledge that a task is difficult or confusing and to encourage students to keep trying, e.g. "Well done. You are working really hard";
- Affirmation prompts on task completion, e.g. "The way that you worked that out was excellent. Now go to the next task. ";
- Instructive feedback – detailed instruction, subtasks, or action to perform to solve the task., e.g. "Use the comparison box to compare your fractions. ";
- Problem-solving feedback, task-dependent feedback – customized feedback per task, e.g. "Good. What do you need to do now, to complete the fraction? ";
- Reflective prompts on task performance and learning – this promotes self-explanation, useful to address learner's misunderstandings, e.g. "What do you notice about the two fractions? ";
- Talk aloud prompts – automatic speech recognition (ASR) has shown to facilitate learning, e.g. "Please explain what you are doing. ";
- Task sequence prompts – provide help on completing the current task when students try to go to the next tasks, e.g. "Are you sure that you have answered the task fully? Please read the task again. ".



(a) Adaptive support components (b) Evaluation of feedback types

Figure 27. Italk2learn – Bayesian techniques to improve learning ITS goals (Grawemeyer et al., 2016)

Evaluation has been performed on a group of 77 participants, split into two groups: 41 users with the adaptive affect condition, and 36 users with the non-affect condition. Both groups are using the iTalk2Learn system, and the evaluation is in the Tutor 1 vs Tutor 2 category, as it was explained in I.2.3. The evaluation included a pre- and post-test, where the pre-tests showed similar results between the two groups. The learning gains on post-tests were not significant, but the overall tendency in the Bayesian affect tutor showed higher learning than the non-affective tutor. Authors showed that the learning experience can be enhanced through feedback tailored to the affective state of the learner. Also, through speech and interaction tutor can automatically deduce their affective states.

5.4.2 Supervised Learning Cognitive Tutors

A Data Mining Driven Cognitive Tutors

Another tutoring system that uses adaptive techniques, called AITS, was developed for school students to enhance their learning gains in AI courses (Grivokostopoulou et al., 2017). The system has an automatic assessment unit with three components: an error detection mechanism, an automatic marking mechanism, and a feedback mechanism. The error detection mechanism identifies any mistakes and categorizes the student's answers based on their accuracy and completeness. The Automatic Marking Mechanism computes the mark for each answer and keeps track of the overall student score. The feedback mechanism provides meaningful feedback related to the score achieved and the errors made per each exercise.

The study employs two categories of artificial techniques: condition-action rule-based reasoning (an expert system based on rules), and data mining techniques (such as analysis based on a decision tree). The techniques were implemented with the goal to present adaptive exercises and predicting student performances. The system estimates the student's level of knowledge and performance. Two experiments were performed for evaluation. In the first experiment, the Pearson correlations¹³ were calculated between the scores given by the tutor and the automated scoring mechanisms. The results were statistically significant and showed positive correlations between the two approaches, as shown in Figure 28. The second experiment evaluated the automated assessment system by dividing the 400 students into five-mark categories: "very low, low, medium, good, and excellent". The results showed that 83% of the students (332 out of 400) were correctly evaluated in the appropriate category of mark level.

Additionally, the study compared the effectiveness of learning in the traditional way versus learning with an artificial intelligence tutoring system (AITS). To do this, 300 undergraduate students from AI classes were split into two groups. There were four phases within the experiment: "pre-test, learning phase, post-test, and questionnaire". The participants in each group were selected randomly, and the results of the pre-test showed no significant differences between ClassA (the control group, comprising 70 female and 80 male students from the academic year 2011-2012) and ClassB (the experimental group, comprising 73 female and 77 male students from the academic year 2012-2013).

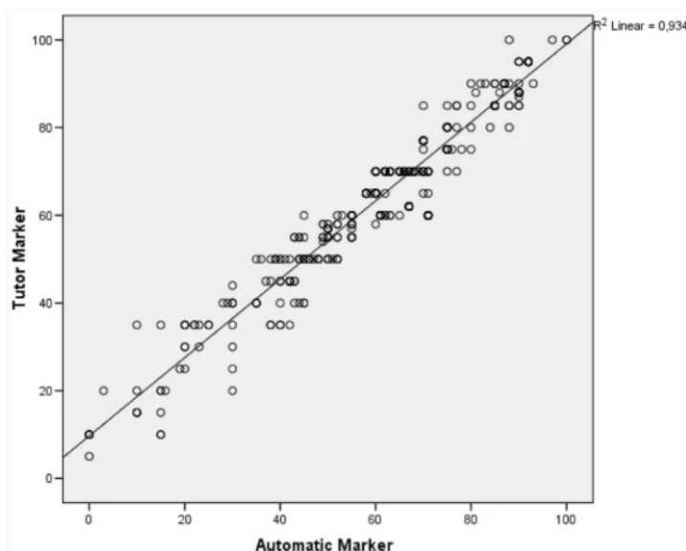


Figure 28. AITS – Scatter plot of Automated Marker vs Tutor Marker (Grivokostopoulou et al., 2017)

¹³ <https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php>

The results indicated that the control group had a mean score of 39.4 on the pre-test and a mean score of 49.98 on the post-test. Meanwhile, the experimental group had a mean score of 38.9 on the pre-test and a mean score of 67.54 on the post-test, which was above that of the control group's score at a significant difference. Additionally, the questionnaire survey found that the automated marker was highly efficient and helpful for tutors.

B Fuzzy-Based Cognitive Tutors

A recent fuzzy-based tutor, called StuDiAsE, has been developed by Samarakou et al. for the diagnosis, assistance, and evaluation of engineering students (Samarakou, Prentakis, Mitsoudis, Karolidis, & Athinaios, 2017). The fuzzy-based model implemented was used for the diagnosis of the student learning profile. Learner's characteristics defined by the tutor include learning style, learner cognitive profile, learner behavior, and skills.

The diagnosis was based on students' responses to questions. Students were also encouraged to engage in the diagnosis process, becoming aware of their learning profile. The questions were classified based on the text comprehension theory, which has the following categories:

- R-text and R-type questions (Relational questions) – related to the hierarchy of part-whole relations in the system;
- M-text and M-type questions (Transformative questions) – related to descriptions of sequences of events and state-to-state transitions in the system;
- T-text and T-type questions (Teleological questions) – related to objectives and sub-objectives of the system, or of other preliminary versions of the system.

The fuzzy model has been validated using the following procedure. Students had an initial diagnostic test with questions in the three categories mentioned above. The first cognitive profile is defined, and personalized feedback is provided further. While learning, the student is required to provide a response to questions according to the materials he is currently studying. The student profile is updated based on his behavior in this phase: the student can skip questions, ask for extra hints, or come back to previous questions. A complete usage log is also stored, including time spent per activity, pages visited, answers changed, or the number of mistakes per question category.

The fuzzification stage included a set of linguistic variables defined as relevant to the recorded data. The output was defined based on the dimensions of the Felder-Silverman learning model (Graf et al., 2006), and included the following output variables: Perception (dimension which includes intuitive or sensing students), Input (dimension which includes visual or verbal students), and Understanding (with students who gain understanding sequentially and students who learn globally).

The inference stage implied the classification of students into categories based on their abilities, a task performed by the research team using 89 type rules (if-then). The results obtained on a group of 28 students are promising, indicating that the fuzzy model can help in deriving meaningful learning profiles. For example, based on the Understanding dimension, students who are perceived more as visual, are ranked by the model in the middle between sequential and global, while the verbal ones are seen to be more sequential and less global. As the authors also mention, the small number of participants involved in the study is a limitation for the current system validation.

Another system using Fuzzy based techniques, Fuzzy rule-based reasoning, called CRITS, has been built to solve computer science programming problems (Mohammed & Mohan, 2015). The objectives of the adaptive techniques used in CRITS were to identify the membership of students in subcultural categories, to assess and update students' knowledge, and to provide adaptive content and feedback. The problem stated by the authors is that the developers of an intelligent tutor create educational content with which they culturally identify. They may alienate or confuse students with different values or perspectives. A culturally neutral tutor is not feasible also, as each student needs a proper learning experience to be engaged.

CRITS is an intelligent tutor that utilizes decision-making which is culturally aware and is initiated by the system, and personalization of educational content. The architecture of CRITS consists of four main components: the Controller, the Cultural Linguistics Adapter, the Contextual Element Repository, and Contextual Student Model. The components with their interactions can be seen in Figure 29. Based on cultural profiles initially defined by the learners, the educational content is adjusted – specific problem descriptions are shown, and specific images, hints, and instructional feedback are provided. Cultural conceptualization involves considering various factors such as language, appearance, and socio-cultural perception in an educational setting. The CRITS organization has identified 24 contextual dimensions that fall into five categories: geographical, ethnic, religious, educational, and physical environment/terrain groups. To understand a student's cultural background, a tutor gathers data and stores it in an ontological structure, using a combination of rules and algorithms to calculate the strength of the student's membership in a particular cultural group. This strength is based on both statistical data and the level of influence of the student's parents.

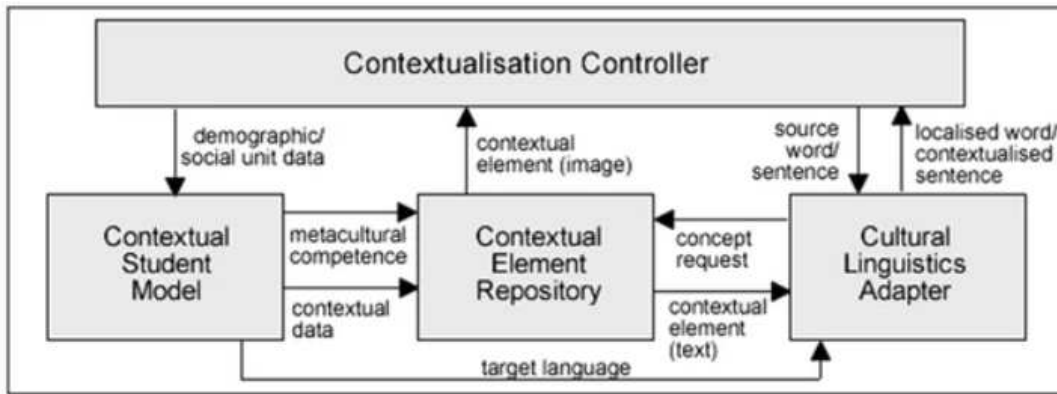


Figure 29. CRITS – Instructional Cultural Contextualized Architecture (Mohammed & Mohan, 2015)

The effectiveness of the CRITS system was assessed based on the user experiences of 33 students taking computer science and IT programming courses at the University of the West Indies. The results of the evaluation showed that there were no statistically significant differences in student ratings based on the type of content or cultural variation, although some interesting trends were observed. The authors found that CRITS was able to adapt to a range of instructional content, including emotive feedback and content descriptions, and could be a useful starting point for incorporating cultural contextualization into intelligent tutoring systems.

5.4.3 Reinforcement Learning Cognitive Tutors

In a recent study on formative assessment as part of tutoring modeling, Melesko and Novickij (Melesko & Novickij, 2019) suggested using the Upper-Confidence Bound (UCB) algorithm for selecting items. Formative assessment is a method of providing feedback on performance in real-time to improve and speed up the acquisition of skills. It involves using UCB, a simpler approach compared to other algorithms such as the Elo rating-based approach and Multidimensional Item Response Theory (IRT), to guide the learning process in real-time.

When a teacher engages in a dialogue with a student to identify gaps in their knowledge, they may need to make decisions about which topics to explore further. For example, for a learning curriculum that covers two subjects, the teacher will equally divide 10 questions for each subject.

Table 16 shows the hypothetical responses of a student, where Topic 1 has two incorrect answers (question 4 and question 10), and Topic 2 has one incorrect answer (question 8). If the teacher gets the opportunity to ask five more questions, the teacher may need to decide which topic to focus on. This becomes more challenging when there are multiple topics and time is limited. In these situations, bandit algorithms can be helpful for balancing exploration and exploitation in the face of uncertainty.

Table 16. Upper-Confidence Bound – Simple double assessment model (Melesko & Novickij, 2019)

Question	1	2	3	4	5	6	7	8	9	10
Topic 1	+		+	-		+				-
Topic 2		+			+		+	-	+	

When selecting the following question to display, the bandit should use the history of all known student answers. The main difficulty in this situation is to identify the topic in which the student is weakest. To overcome this challenge, the bandit should tackle various subjects and, at the same time, exploit other topics to get a more exact estimation of students' knowledge of the topic. Evaluation has been performed in both simulated environments and with real students, 104 undergraduate, third-year students, from Vilnius Gediminas University, Lithuania. The data collected for each student was based on their responses to a quiz with 60 questions on 15 subtopics, such as network topologies, networking devices, Ethernet, and cloud computing services. The UCB algorithm demonstrated better performance in terms of exploration compared to randomly selecting questions for every factor tested.

The results of the research demonstrated that if the goal is to accurately identify the weakest topic for each student with a 95% accuracy rate, traditional testing methods could be shortened from 60 questions to 55 questions. However, using the UCB approach with a multi-armed bandit algorithm, the quiz length could be reduced to 32 questions while still achieving 95% accuracy, which is a significant improvement. Results can be seen in Figure 30. The simulated UCB is plotted with a gray line, and it has shown that the synthetic experiments are congruent with the real testing.

The study demonstrated that UCB algorithms have the prospective to considerably reduce the length of assessments without sacrificing accuracy. The results were particularly strong for tests with fewer topics, and this approach could be applied in both traditional classrooms and ITSs to improve their effectiveness.

Another intelligent system based on the MAB algorithms for personalized assignment of learning tasks to students has been proposed by Andersen et al. (Andersen, Kråkevik, Goodwin, & Yazidi, 2016). The study introduced a new algorithm called the Skill-Based Task Selector (SBTS), based on multi-armed bandit algorithms and utilizing two types of data in tutoring: the topics that students should focus on next and the level of difficulty of each task.

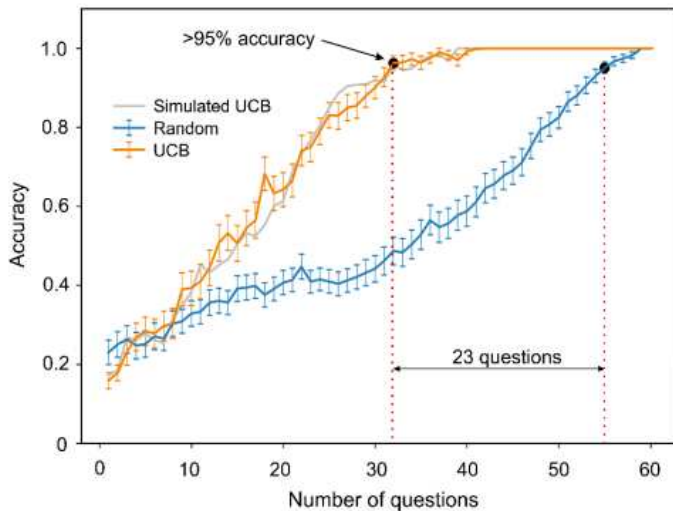


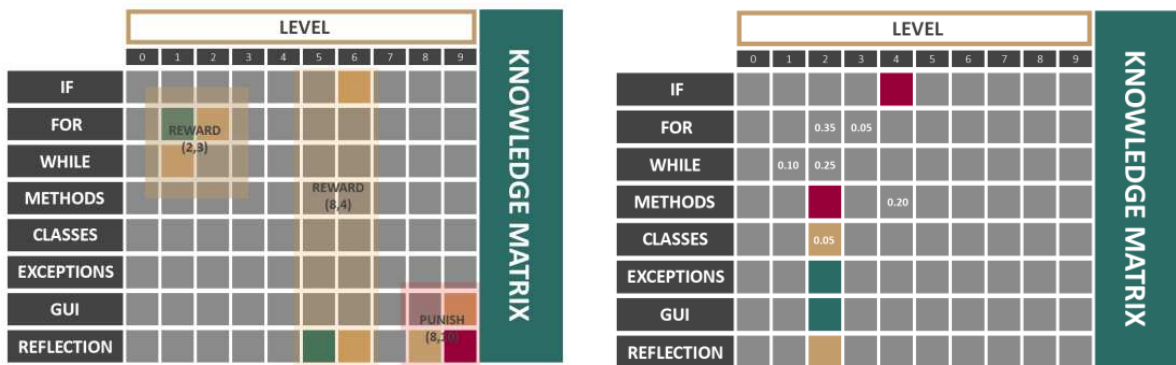
Figure 30. Average accuracy of the formative assessment tutoring (Melesko & Novickij, 2019)

The SBTS algorithm adapts through reward and punishment schemes and maintains a skill matrix that is updated considering the learner's behavior. The goal of the system is to offer individually customized learning in computer programming courses. The skill matrix in this experiment consists of eight topics, represented as rows in the matrix: "if, for, while, methods, classes, exceptions, GUI, and reflection". The columns represent the level of difficulty for each topic, with ten levels per topic. The topics are ordered in the matrix by difficulty, from if to reflection. The algorithm does not have any prior knowledge of the level of difficulty per topic. Every matrix cell has a value showing the relevance of that task level for the learner and indicated as a probability. A higher value rises the probability of that task level being chosen. The sum of all the cells in the matrix is always 1. The punishment and reward functions update the cell values in the matrix.

An example of a skill matrix calculation can be seen in Figure 31. Figure 31(a) shows that the reward pushes the agent to move "down and right" while the punishment pushes them "up and left." In Figure 31(b), the area highlighted with colors represents the expected knowledge domain according to the present student aptitude level. Green cells are the best option, and since the skill may be inaccurately estimated, the algorithm should choose between green cells and high probabilities from the matrix. The evaluation of the approach has been performed in a simulated environment with 1000 students for 200 task sets with 1000 iterations. Two study cases have been performed, with two approaches of the epsilon-greedy algorithm, one with static epsilon, and one with dynamic epsilon. It has been demonstrated through simulation that the static epsilon approach was 30% better than the dynamic epsilon approach. The authors conclude their proposal is ready to be tested and validated through practical trials in classroom settings.

Another tutor which uses Multi-Armed Bandit techniques for personalizing the sequence of learning activities has been proposed by Clement et al. (Clement et al., 2015). The system is evaluated with

400 schoolchildren, 7 and 8 years old, on number decomposition while manipulating money. Also, an initial system evaluation is performed in several simulated experiments. The authors of the study addressed the issue of using mathematical concepts related to quantities and monetary values as a means of applying abstract knowledge to practical, real-life scenarios that can be beneficial for children.



(a) Visual representation of punishment and reward (b) Skill level calculation

Figure 31. Knowledge matrix and update matrix rules (Andersen et al., 2016)

Teaching instruction includes seven Knowledge Components (KC): "(a) KnowMoney (capability to handle money to buy objects), (b) SumInteger (capability to add integers), (c) SubInteger (capability to subtract integers), (d) DecomposeInteger (capability to decompose integers into groups of ten and units, (e) SumCents (capability to add decimal numbers), (f) SubCents (capability to subtract decimal numbers, (g) DecomposeCents (capability to decompose decimal numbers)" (Clement et al., 2015).

The authors of the study proposed two algorithms for adaptive learning sequences: ZPDES (Zone of Proximal Development and Empirical Success) and RiARiT (Right Exercise at the Right Time). ZPDES doesn't utilize explicit information regarding the learners, and it relies solely on their past learning history, while RiARiT explicitly estimates the learner's level through a method resembling Bayesian knowledge tracing. In the multi-armed bandit implementation, the choice of arms corresponds to the chosen learning activities, and the reward is the student's learning progress. The assumption is that activities with a higher learning gain should be selected more often. However, a learning activity will no longer provide a positive reward or learning progress once the student reaches a specific level of competence. Additionally, the same exercise may provide different rewards for two students with similar knowledge levels due to their individual characteristics.

The ZPDES algorithm is relatively simple and requires minimal domain knowledge. It describes a relationship between the learning activities, based on incremental difficulty levels, and the exploration/exploitation problem is driven by the exploration graphs defined by the authors. RiARiT algorithm defines a relation between Knowledge Components (KC) and Pedagogical Activities. Students are modeled with a competence level across all KC, which is a value between 0 – KC not acquired and 1 – KC fully acquired. RiARiT, like ZPDES, establishes a set of minimum competence levels that a student must achieve in order to access certain learning activities.

For evaluation, the 400 students have been split into 4 groups: one control group (without the software), one group with Expert Sequence, one group using the ZPDES algorithm, and one group with the RiARiT algorithm. In this study that included pre- and post-tests, all groups participated and the results showed that ZPDES generally performed better than the other algorithms in the real experiments, as indicated in Table 17. Table 17. System Evaluation. Average knowledge acquired per type of algorithm (Clement et al., 2015)

	Level Average			
	M	R	MM	RM
ExpertSeq	5.42	1.66	1.41	0.14
RiARiT	4.47	1.74	1.77	0.70
ZPDES	2.79	2.01	1.83	1.05

M, R, MM, and RM are exercise types, defined in incremental difficulty order. The authors conducted two statistical tests to further evaluate the efficacy of the ZPDES approach, and the results showed that it is effective. One of the advantages of ZPDES is that it requires very little information and has fewer parameters compared to other algorithms.

Based on the above considerations and the similarities between Clement et al. (2015) approach and the tutoring athleticization in open environments, we will further propose a reinforcement learning algorithm for psychomotor training, which will be described in Chapter 8 and the results obtained will be presented in Chapter 9.

Experiments and Results

6 Experiments and Results

Chapters 5-7 present our modeling, which involves knowledge, tutoring, and communication modeling. Chapter 8 describes the experiment which was conducted to validate our modeling and the results obtained. The model proposed and the validation has been performed following our findings described in Theoretical approach.

Chapter 5 introduces a knowledge model in the psychomotor field, called *OntoStrength*, built using an ontological approach, with a focus on strength skill development. The methodology used and each implementation step are described in this chapter. Its structure is also presented, which includes *Strength Skills*, *Strength Development Program*, and *Personalized Development for Strength Skills* sub-domains. Finally, we partially present content within *OntoStrength* and methods to integrate such a knowledgebase with intelligent tutoring systems.

Chapter 6 introduces a tutoring model based on a reinforcement learning approach, using a contextual multi-armed bandits algorithm. The proposed model, called *RiERiT* – Right Exercise at the Right time, is described, to improve the learning instruction on psychomotor training. Context variables are defined, and multi-armed bandits' implementation is presented. Last, simulations are performed with variations of bandits' algorithms, and the results obtained conclude the current chapter.

Chapter 7 presents a prototype of an Intelligent Tutoring System in Psychomotor training, called *Selfit*. The prototype description includes an architectural overview, and a presentation of each module – student, domain, tutoring, and interface. *Selfit* uses the previously introduced ontology - *OntoStrength* - for the domain, student, and training rules in the tutoring module. Also, the tutoring is enabled through *RiERiT*, introduced in the previous chapter. In this chapter, the interface module that enables communication between the learner and the system is described and several examples of screens that facilitate this communication are presented.

Chapter 8 focuses on describing an experiment where the *Selfit* prototype is tested with real users. The goals were to validate the knowledge base created, the tutoring validity, and if the interface module is efficient and pleasant to the users. Two versions of the system were tested at different times – version 1, with 18 trainees, and version 2, with 42 trainees – and the results obtained are presented.

7 OntoStrength – A Framework to Represent and Inference Knowledge in Psychomotor Intelligent Tutoring System

7.1 Résumé

Ce chapitre présente OntoStrength, une base de connaissances développée par une équipe d'informaticiens et de scientifiques du sport pour soutenir le développement d'un STI psychomoteur. L'ontologie couvre le domaine du développement de la force et comprend des informations sur les compétences en matière de force, les processus de développement de la force et les variables de développement personnalisées. La méthodologie *Ontology Development 101* a été utilisée pour développer l'ontologie, qui consiste en 618 classes et relations entre elles. L'ontologie a été éditée et affinée à l'aide du logiciel Protégé et stockée dans une base de données GraphDB.

Le sous-domaine des aptitudes à la force comprend des descriptions des aptitudes à la force, y compris les forces générales et spécifiques, les aptitudes au mouvement et les contractions des muscles humains. Le sous-domaine du processus de développement de la force couvre les étapes du développement de la force. Le sous-domaine des variables du développement personnalisé couvre les caractéristiques individuelles qui doivent être prises en compte pour les tâches du programme de musculation personnalisé.

OntoStrength utilise SPARQL pour soutenir la création de programmes de développement de la force. Il utilise une hiérarchie de classes pour représenter les différents niveaux d'entraînement (macro-cycle, mésocycle et micro-cycle) et leurs relations. Les séances d'entraînement sont créées selon les règles de Bompa, et chaque exercice est décrit en termes de mouvements fonctionnels et fondamentaux, et de contractions musculaires impliquées. Les résultats d'une requête SPARQL peuvent être affichés dans différents formats et la hiérarchie d'héritage peut être visualisée dans GraphDB, montrant comment chaque exercice est un individu nommé d'une classe avec des caractéristiques spécifiques (telles que le type de mouvement, les mouvements fonctionnels et les contractions musculaires impliquées).

La dernière section de ce chapitre présente l'utilisation d'OntoStrength. L'ontologie est accessible via une API RESTful, qui fournit des microservices à des fins d'entraînement sportif. Ces microservices peuvent être intégrés dans n'importe quel STI pour le développement psychomoteur. L'ontologie sert de base pour générer des chaînes d'entraînement, y compris des séances d'entraînement, des microcycles, des mésocycles et des macrocycles. L'ontologie sera publiée en tant que source ouverte

et pourra être étendue à d'autres capacités psychomotrices telles que la flexibilité, la mobilité et les capacités d'endurance.

7.2 Introduction

The current chapter introduces OntoStrength, an Ontology for psychomotor skill development, more specifically strength development, and its integration within a Psychomotor Intelligent Tutoring System.

The chapter is organized as follows: First, we introduce similar work in the ontology field, as well as examples of knowledge representation practices in ITSs; together with solutions developed specifically for the psychomotor domain. Afterward, we present the OntoStrength ontology with the aim to organize the sports strength training knowledge. The last two sections illustrate its usage and potential future research directions.

7.3 OntoStrength's Design Methodology

This section presents an overview of OntoStrength, the underlying engineering process, as well as details on the ontology. After presenting its structure, three subdomains are described: strength skills, strength skills development process, and personalized development variables.

7.3.1 Design Principles

The OntoStrength ontology aims to support the development of a psychomotor ITS dedicated to enhancing psychomotor skills and associated bio-motor abilities, such as strength. A multidisciplinary team composed of computer scientists and sports scientists developed OntoStrength. Based on the findings from II.1 4.3.1A, the Ontology 101 methodology, also known as OD101, (Noy & McGuinness, 2001) was used to develop the ontology.

The first step in this process involves defining the domain and scope of the ontology through the formalization of competency questions. We characterized competency questions by considering student requirements, as well as the domain and tutoring models from the ITS. The first set of competency questions raises the need for the ontology to cover the diversity of strength skills. Then, a second set considered the complexity of strength development. Finally, the third set of questions raised the need to model variables supporting the personalization of development processes for each student.

Then, we performed the second step of the methodology design, consisting of the identification of existing ontologies. Typologies about muscles, strength development exercises, training planification rules, or injuries were collected. No similar ontology was identified in the literature for these fields.

The third step of the methodology engineering process consisted of the identification of essential terms for the domain. The team reviewed empirical literature about strength training to compile the first list of terms to be considered by the ontology.

Steps four, five, and six were focused on defining classes, class hierarchies, and associated slots and facets; as such, 618 classes were created in OntoStrength. The last phase was centered on the instantiation of entities or individuals. This group of tasks was iteratively performed several times. The refinement of the ontology was based on expert validation and experiment sessions.

We used the Protégé software to edit and refine classes, relationships, slots, and facets. GraphDB¹⁴ was considered the semantic graph database for storing and querying the ontology, as well as for generating interactive data plots. The SPARQL query language (Prud'hommeaux & Seaborne, 2008) was used to test different training scenarios through queries.

7.3.2 OntoStrength Structure

Strength development is the domain covered by OntoStrength. Strength is defined as the maximum force or torque that a muscle or muscle group is capable of producing, or as the ability of the neuromuscular system to exert force against an external resistance (Bompa, 2017).

Strength development includes a set of essential tasks, such as the assessment of student strength skills, the definition of a development program, and the monitoring of its application. The ontology provides knowledge to support the functionalities of a psychomotor ITS.

First, OntoStrength supports the ITS domain module by providing classes that describe the diversity of strength skills. Second, classes on strength development processes support the tutoring module. Finally, OntoStrength supports the ITS student module with knowledge about the different individual characteristics to consider for the personalization of strength program tasks.

¹⁴ <https://graphdb.ontotext.com/>

7.3.3 OntoStrength Sub-domains

A *Strength Skills Sub-domain*

The first OntoStrength sub-domain aims to describe strength skills. The domain module uses strength skills to provide development objectives, the student module uses it to design a student strength fingerprint, while the tutoring module generates and monitors training workouts as an input. When defining strength, Bompa (Bompa, 2017) makes the distinction between general and specific strengths. The first type of strength refers to the overall strength of the muscular system, while the second is related to the muscle group patterns that are important for a specific activity. He also defines three strength properties: power, maximum strength, and strength endurance. Power is the ability to develop force rapidly and at high velocities. Maximum strength refers to the highest force the neuromuscular system can generate during a maximum voluntary contraction. Finally, strength endurance is the ability of the neuromuscular system to produce force repetitively over extended periods.

Hence, a strength skill combines a movement skill and a strength type in the OntoStrength ontology. OntoStrength considers four movement types: Muscular, Functional, Fundamental, and Specialized. The first two movements support the general strength skills. *Muscular* describes the different types of possible contractions for each muscle involved in human body movement. OntoStrength describes twenty-four muscles and four contraction modes: eccentric, concentric, isometric, and plyometric. *Functional* is related to actions performed by the human body joints while moving.

The ontology includes ten joints (two for each hip, elbow, knee, neck, and shoulder) and associated actions (such as abduction, rotation, extension, and flexion). The fundamental and specialized movement types support the description of specific strength skills. *Fundamental* patterns are "the building blocks that lead to specialized movement sequences required for adequate participation in organized and non-organized physical activities" (Goodway, Ozmun, & Gallahue, 2019). They include stability, locomotor, and manipulative motor patterns. *Specialized* movement skills include daily life, leisure, sports, and motor patterns specific to professionals.

The OntoStrength strength skills sub-domain includes three hierarchies of classes that can be used to describe skills such as "Biceps Eccentric Maximum Strength", "Hip Flexion Strength Endurance", "Throwing Maximum Strength", or "Lunge Power". The classes used to represent the main concepts of this domain are the following: "Strength Skill", "Movement Skill", and "Strength Property". The architecture of the OntoStrength strength skills sub-domain can be observed in Figure 32.

A "Strength Skill" instance is composed of an instance of "Movement Skill" and an instance of "Strength Property". Even though the OntoStrength ontology aims to support a digital solution for strength development, the design process strived to ensure extensibility to the other bio-motor skills and other usages than strength development, such as risk prevention.

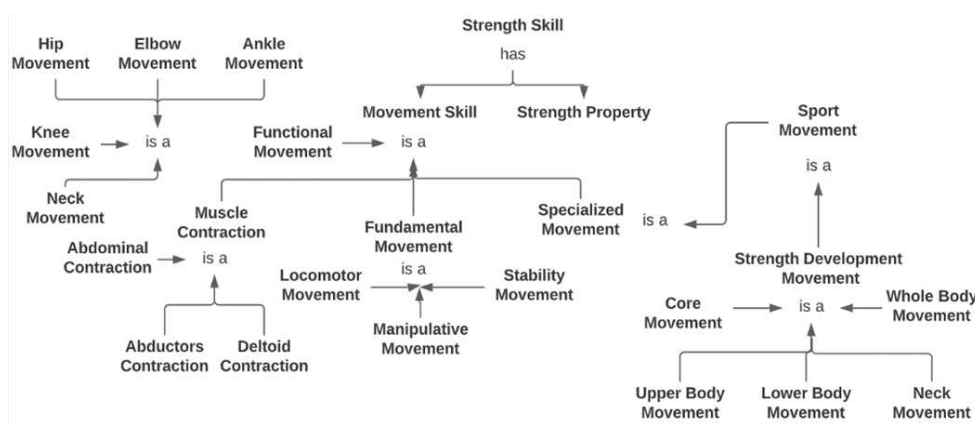


Figure 32. OntoStrength sub-domain for Strength Skills.

The first three subclasses of "Movement Skill": (i.e., "Muscle Contraction", "Functional Movement", and "Fundamental Movement") are independent of any context and application and can be used to describe any movement. The "Specialized Movement" subclass focuses on movements used to develop strength skills. Moreover, this subclass provides the potential of reusing the ontology for any other movement skills and applications.

The "Muscle Contraction" class has twenty-three subclasses related to each essential muscle involved in human movements. For example, one of the classes is the "Biceps Contraction Class" with four subclasses: "Biceps Concentric Contraction", "Biceps Eccentric Contraction", "Biceps Isometric Contraction", and "Biceps Isometric Contraction". The "Functional Movement" class has ten subclasses related to each joint movement. For example, one of the classes is the "Knee Movement" class with four subclasses: "Knee Extension", "Knee Flexion", "Knee Lateral Rotation", and "Knee Medial Rotation" classes.

The "Fundamental Movement" class has three subclasses, each with specialized movements. For example, the "Locomotor Movement" class has associated the following subclasses: "Bounding", "Dodging", "Gallop", and "Jumping". The "Specialized Movement" class hierarchy supports the description of all movements associated with activities. They are structured by considering Sports, Professional, Leisure, and Daily Life categories. The OntoStrength ontology focuses more on a subclass of the "Sport Movement" class, namely the "Strength Development Movement" class used to develop strength skills. These skills are structured into five types: "Core Movement", "Lower Body Movement", "Neck Movement", "Upper Body Movement", and "Whole Body Movement".

Each of them has sub-classes, for example, "Anti-Extension Movement" and "Anti-Flexion Movement" for the "Core Movement". Finally, classes related to exercise patterns were also represented – for example, the "Bird Dog" class is a "Hip Dominant Movement".

OntoStrength characterizes a strength development movement task with four attributes: a description of the movement, a typology of the movement (basic, ballistic, complementary) for selecting exercises when planning a workout, muscles targeted by its use, and joint movement actions involved. For example, an instance of the "Diamond Triceps Push Up" class is a basic task for developing "Triceps Brachii", "Pectoralis Major", and "Anterior Deltoid" by performing "Shoulder Transverse Flexion", "Shoulder Flexion", "Shoulder Girdle Protraction", and "Elbow Extension".

B Strength Development Program Sub-domain

The second OntoStrength sub-domain supports the description of a strength development program. The domain module for a psychomotor ITS can use this knowledge to provide relevant content to generate and schedule training workouts. The student module uses this knowledge to update student training components when performing workouts. Finally, the classes structure the behavior of the tutoring module.

Improving strength involves a systematic process in which the body and mind are subjected to varying levels of stressors in terms of volume and intensity (Bompa, 2017). The implementation of a strength development program starts by defining the temporal units that structure the different phases of the strength development process. Depending on its duration, macro-cycles (six to twelve weeks), mesocycles (two to six weeks), micro cycles (five to ten days), and workouts (thirty minutes to two hours) are used to generate the development program.

Issurin (2015) considers three types of mesocycle (i.e., Accumulation, Transmutation, and Realization), six types of micro-cycles (i.e., Adjustment, Competitive, Impact, Loading, Precompetitive, and Restoration), and seven types of blocks of tasks (i.e., Warmup, Combined, Conditioning, Exams, Tactical, Technical, and Cooldown). Afterward, training goals and content are defined for each temporal phase.

All strength training programs start with an initial preparation phase, named anatomical adaptation, which aims to get the body in a good shape and learn the exercise execution; the content and duration of this phase depend on the learner's experience. According to the learner's needs and objectives, goals can be either hypertrophy (i.e., an increase in muscular volume) or the development of maximal strength, power, or endurance. Exercises are then selected, and preliminary assessment tests are conducted to estimate the learner's level for each strength skill. Finally, the workout content is

defined. The content is composed of a set of exercises, with associated duration, intensity, tempo, and rest time. These variables depend on the session goal, the learner's overall level, and his/her current shape, as well as the content from previous sessions. A full training history supports the monitoring of the development process and, if necessary, its refinement.

Hence, this sub-domain centered on strength skill development includes two hierarchies of classes: one to describe different periods and one to represent strength development modalities. A simplified view of the subdomain can be seen in Figure 33.

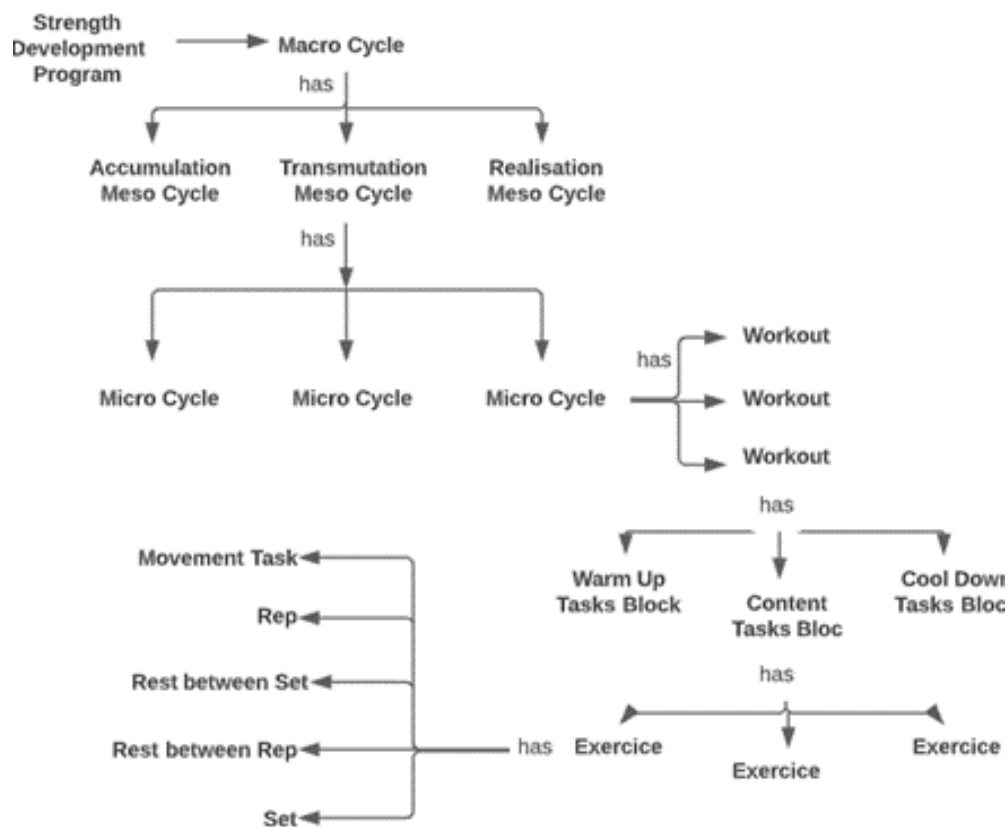


Figure 33. OntoStrength sub-domain for Strength Development Program

The "Biomotor Development Period" class hierarchy consists of all possible temporal periods. Each period is characterized by temporal information about the beginning and the end, coupled with objectives. Furthermore, the "Macro Cycle" class is composed of three instances of the "Meso Cycle" class hierarchy: one of the "Accumulation Meso Cycle" class, one of the "Transmutation Meso Cycle" class, and one of the "Realization Meso Cycle" class.

Each mesocycle is composed of two to six instances of the "Micro Cycle" class. Each "Micro Cycle" instance is composed of one to ten instances of the "Workout" class. An instance of the "Workout" class is made up of three instances of the "Tasks Block" class: an instance of the "Warm Up Block" class, an instance of the "Content Block" class, and an instance of the "Cool Down Block" class. An

instance of the "Tasks Block" class is characterized by a ranked list of instances of the "Exercise" class. An instance of the "Exercise" class includes an instance of the "Movement Task" class and an instance of the "Load" class. Instances of the "Load" class are characterized by the number of sets, the number of repetitions, the rest time duration between two sets, and the rest time duration between two repetitions.

The "Training Target" class hierarchy consists of all possible training modalities integrated into the ontology. The "Strength Training Target" class hierarchy is related to the four strength training modalities represented by the classes "Anatomical Adaptation Target", "Hypertrophy Target", "Maximal Strength Target", and "Endurance Strength Target".

C Personalized Development for Strength Skills Sub-domain

The third OntoStrength sub-domain supports the description of variables used when defining strength training development programs adjusted to students' characteristics. The student module from a psychomotor ITS uses this knowledge to provide the tutoring module with specific knowledge about each student. In addition, the tutoring module uses this knowledge when defining workout content and updates it based on the feedback received from the student. A typology of strength fingerprints structures these variables.

The "General Signature" class contains generic attributes such as the student's name, age, gender, size, or weight. One specific signature is associated with each bio-motor ability. The "Strength Signature" includes all strength skills relevant to each student development program.

The "Anthropometric Signature" refers to body sizes, weight, and body composition. The "Injury Signature" includes each student's history of relevant injuries to be considered when performing a strength development program. The "Motor Signature" associates specific levels to the student, for each movement skill. Moreover, the ontology includes the level of each strength movement type for strength development.

The "Strength Training Signature" describes the student training history for each workout performed, the content, success evaluation, and associated student feedback. Hence, the "Personalized Biomotor Development variables" organize all the different signatures in the OntoStrength ontology, as can be seen in Figure 34.

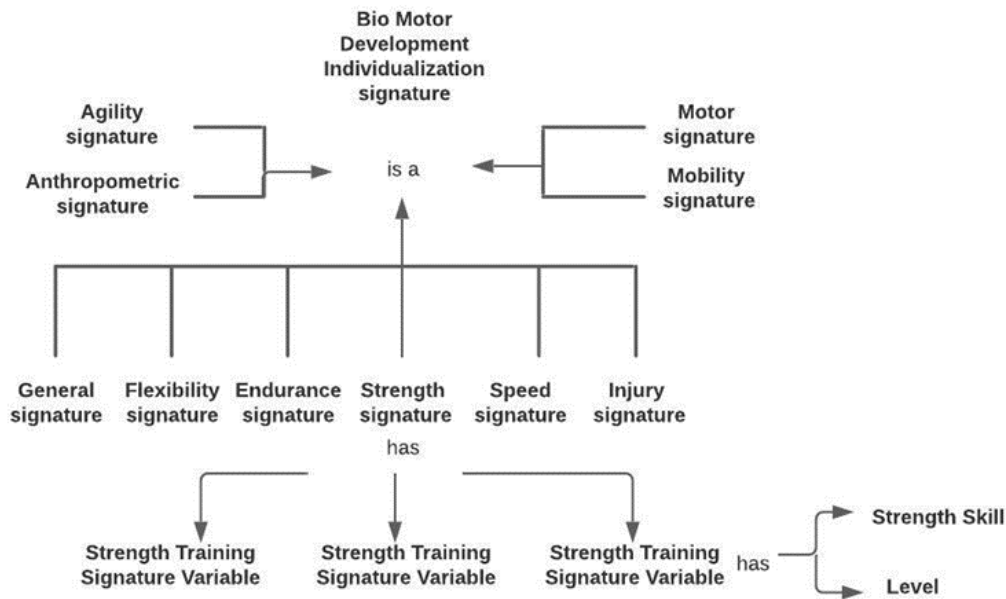


Figure 34. OntoStrength sub-domain for Personalized Development of Strength Skills

7.3.4 OntoStrength Content

The previously described major class hierarchies support the instantiation of strength development programs, from the macro-cycle level to the exercise level. In addition, SPARQL queries were implemented to solve specific training tasks – for example, to obtain exercises associated with a specific body part (wide triceps push-up, side to side pull-up, feet elevated pike push-up), or to obtain generic training templates for a training objective, based on trainee characteristics. GraphDB was used to test the queries and interact with the ontology. Through SPARQL, new data can also be added to the ontology.

OntoStrength presents the instantiation of a Macrocycle entity, which has a Mesocycle entity as an object property, called `hasMesoCycle`. The Mesocycle entity has a Microcycle entity, as an object property, called `hasMicroCycle`. The Microcycle has two object properties (`hasWorkout` and `hasObjective`), whereas the Microcycle is defined as a Push Workout and a Full Body Workout.

The workouts are initialized with exercises following the rules defined by Bompa (Bompa, 2017), for a beginner-level load. Each exercise instance describes a list of functional and fundamental movements, together with the muscle contractions involved in the execution. The SPARQL query from Figure 35 retrieves all movements involved in a specific Microcycle defined in OntoStrength ("os" denotes the prefix specific to the OntoStrength ontology).

```
SELECT DISTINCT ?movement ?microCycle
WHERE
  ?microC os:hasMicroCycle ?microCycle.
  ?wkout os:hasWorkout ?workout.
```

```

?cntBlck os:hasContentBlock ?contentBlock.
?ex os:hasExercise ?exercise.
?ld os:hasLoad ?load.
?mvmnt os:hasMovement ?movement.

```

Figure 35. OntoStrength MicroCycle Movements query.

The query was applied in the GraphDB web interface, and it returns 11 unique movements, such as Archer Push Up, Chest Dip, or Feet Elevated Pike Push Up, which are applied as composed movements in all workout exercises.

GraphDB has many options to display the results of a SPARQL query ranging from the Table view, Raw Response – JSON formatted data, and Pivot Table, to more advanced Google Charts. All unique functional movements for the defined macrocycle MicroCycleAAPDupont_001 are displayed in Figure 36.

movement	microCycle
#ArcherPushUp	#MicroCycleAAPDupont_001
#ChestDip	#MicroCycleAAPDupont_001
#PushBack	#MicroCycleAAPDupont_001
#FeetElevatedFrontPlank	#MicroCycleAAPDupont_001
#Superman	#MicroCycleAAPDupont_001
#BirdDog	#MicroCycleAAPDupont_001
#FullSquat	#MicroCycleAAPDupont_001
#FeetElevatedInvertedRow	#MicroCycleAAPDupont_001
#BicepsChinUp	#MicroCycleAAPDupont_001
#FeetElevatedPikePushUps	#MicroCycleAAPDupont_001
#FrontPlank	#MicroCycleAAPDupont_001

Figure 36. GraphDB Google Chart SPARQL Query Results for Unique Functional Movements to train in a MicroCycle

The inheritance hierarchy can be also visualized in GraphDB. OntoStrength relies on property inheritance between classes. Each exercise is represented as a class, which has specific base classes representing the type of movement involved, while the most generic one is the "Movement Skill" class. The "Exercise" class is inherited from the "Specialized Movement" class, which includes a list of "Functional Movements" to execute (such as "Elbow Extension" or "Shoulder Adduction"). Moreover, "Functional Movements" are composed of a list of "Muscle Contractions", such as "Hamstrings Concentric Contraction", "Quadriceps Eccentric Contraction", and "Teres Major Isometric Contraction".

An instantiation of an "Exercise" class is the exercise itself, with its specific level and description. "Feet Elevated Front Plank" from Figure 37 is an example of a named individual of the class with the same name, described as a Level 1 exercise, for the Anti-Extension Movement.

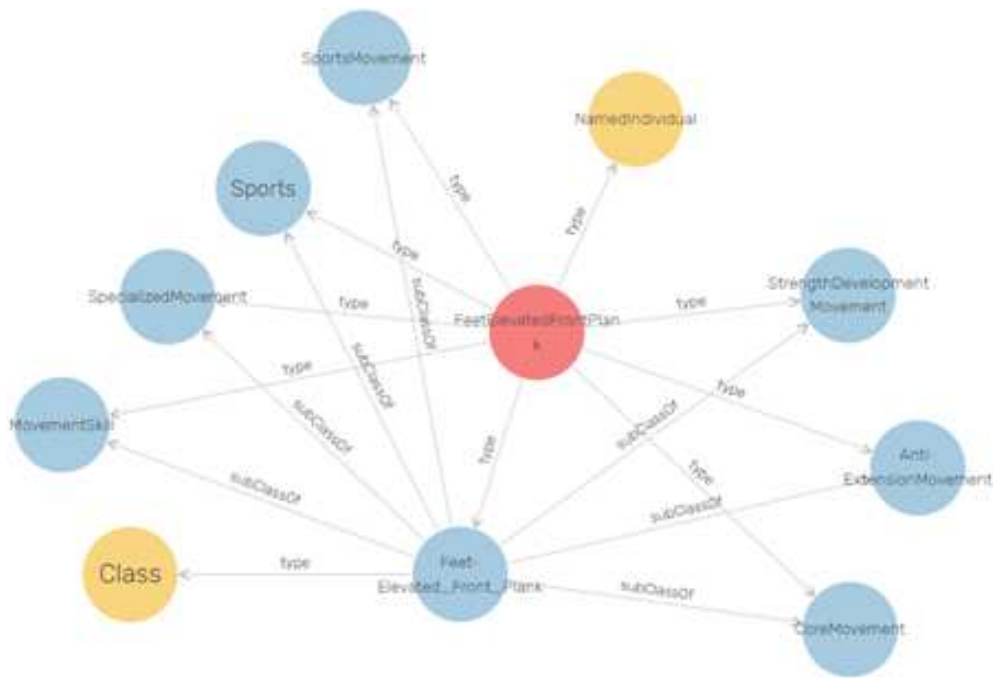


Figure 37. GraphDB View of Feet Elevated Front Plank instance and class

7.4 OntoStrength Usage

The OntoStrength ontology supports any psychomotor ITS by providing knowledge and relationships useful to different modules of the system. To this end, a RESTful API can interact with the ITS to provide microservices specific to the domain, student, and tutoring system functionalities. The following subsections illustrate the interactions of OntoStrength with a psychomotor ITS via the RESTful API.

The ITS Student profiling functionalities elaborate and refine student signature profiles used to personalize the definition of strength development workouts. The OntoStrength RESTful API provides queries to support the selection of signature variables used by the ITS, update signature variables, and obtain the values of signature variables, as seen in Table 18.

The strength individualization signature consists of a level ranging between 1-4 to match the level at which exercises can be safely performed, for all strength movement skills. The evaluation of these parameters is performed through an initial calibration workout session, which has an incremental complexity, until failure.

For example, "Side to Side Push Up" for upper body area level 1 complexity or "Self-Assisted One Arm Push Up" for level 2. This system calibration feature uses the RESTful API to obtain the exercises for a given strength movement skill and level, to get the actual student level for a selected strength movement skill and update the student profile after each exercise. Once an initial student

calibration is performed, a strength development training program is generated and is continuously updated after each workout.

Table 18. OntoStrength RESTful API student's profiling function

Query	Result	Input	Output
getSignatureList	Return all signatures associated with a student profile	Student	List of signatures
getSignature	Return details of a specific signature associated with a student profile	Student, Signature	Signature details
getSignatureVariables	Return the variables used to provide personalized development workouts for a given strength skill	Strength skill	List of signature variables
getSignatureValueForVariable	Return the value of a student signature variable	Student, signature variable	Value of the signature variable
setSignatureValueForVariable	Update the value of a student signature variable	Student, signature variable, new value	Status to confirm the update
getMovements	Return a list of movements	Movement class, level	List of movements

This work presents the design and development processes of an ontology dedicated to strength psychomotor capacity called OntoStrength. As per our knowledge, although there is an increasing demand for building digitalization solutions in the healthcare and sports training fields, there is no previous work performed on the psychomotor domain.

We considered Ontology Development 101 (OD101) for building OntoStrength, a micro-level methodology that proposes a practical and explicit guide for developing ontologies. The process, based on interdisciplinary teamwork, implied a clear definition of the domain and the scope, the reuse of existing ontologies, the definition of classes, and properties, and finally instance creation to complete the knowledge base.

The ontology can be accessed using a microservices architecture, where specific endpoints are available to serve multiple queries for sports training purposes, such as exercises targeting a movement pattern, by difficulty, warm-up sessions generation, weekly training generic plans, based on objectives, and others. The microservices are ready to be integrated into any ITS for psychomotor development, where core components (i.e., student, domain, and tutoring models) may rely on the knowledge base for conceptualization and basic inferences.

The current section is the foundation to generate entire training chains, including workouts, micro cycles, mesocycles, and macrocycles. Improvements regarding the design of the ontology include additional gender dimensions and assessments of injuries while practicing training activities. OntoStrength will be released under an open-source license and will be expanded to include other psycho-motor abilities such as flexibility, mobility, and endurance skills.

8 Teaching Strategies in a Psychomotor Tutoring System

8.1 Résumé

Cette partie traite de l'utilisation d'un modèle de tutorat qui incorpore un algorithme contextuel de bandit à bras multiples pour personnaliser l'entraînement sportif. L'apprentissage par renforcement est utilisé pour surmonter les difficultés liées à la préparation de l'entraînement et pour créer une expérience d'entraînement plus efficace. Le modèle tient également compte des différences physiologiques entre les hommes et les femmes en intégrant le dimorphisme sexuel dans sa conception. Le système a été évalué dans un environnement virtuel et a montré que la prise en compte du dimorphisme sexuel est importante pour l'entraînement personnalisé et la prévention des blessures.

Il aborde ensuite le dimorphisme sexuel dans l'entraînement psychomoteur. Les différences de caractéristiques morphologiques, cognitives et physiologiques entre les hommes et les femmes peuvent avoir un impact sur la conception des séances d'entraînement et augmenter le risque de blessures et de troubles psychologiques. Les femmes ont une structure squelettique plus petite et une plus grande quantité de graisse corporelle, une capacité aérobie plus faible et moins de force dans le haut du corps que les hommes. Les femmes ont tendance à être plus performantes dans les tâches de reconnaissance verbale, de fluidité sémantique et d'habileté manuelle fine, tandis que les hommes ont tendance à être plus performants dans les tâches de navigation spatiale, les tâches sensorimotrices et la rotation mentale. Le cycle menstruel et les changements hormonaux qui y sont associés peuvent affecter les performances sportives d'une femme et augmenter le risque de blessure au LCA. En outre, les troubles alimentaires, l'ostéoporose et l'aménorrhée peuvent également augmenter le risque de blessure chez les athlètes féminines. Lors de la programmation de séances d'entraînement pour les femmes, il faut tenir compte de facteurs tels que le cycle menstruel et les changements hormonaux associés aux menstruations. La charge de la session peut être ajustée en fonction des performances de la stagiaire, et une semaine de repos peut être programmée toutes les trois semaines pour faciliter l'apprentissage et éviter la sur-fatigue.

La dernière section présente le modèle de tutorat proposé dans la présente thèse, appelé méthode Right Exercise at the Right Time (RiERiT). Le processus de tutorat pour notre STI psychomoteur est structuré sur une échelle de maturité à quatre niveaux qui comprend des formateurs novices, intermédiaires, avancés et experts. Les tuteurs utilisent l'algorithme des bandits à plusieurs branches pour personnaliser les séquences d'entraînement, les sessions, les micro-cycles, les mésocycles et le

contenu des macro-cycles. Le modèle utilise des modèles de séquences d'entraînement et se base sur les données fournies par le stagiaire, telles que le nombre de sessions, la durée de l'entraînement et les objectifs. Les exercices sont sélectionnés en fonction des progrès du stagiaire estimés par des défis d'étalonnage, et l'algorithme utilisé est celui des bandits contextuels à bras multiples. La récompense de l'algorithme d'apprentissage par renforcement est calculée comme la différence entre la charge externe et la charge interne, qui est représentée par les répétitions en réserve (RiR) et dont les valeurs se situent dans l'intervalle $[-10, 10]$. La formule de récompense et les descriptions de RiR concluent le présent chapitre.

8.2 Introduction

Proper training planning and execution at the highest level are crucial for success in sports (Me & Unold, 2011). One common challenge in training preparation is determining the proper stimulus for each athlete at the appropriate time. There are very few studies that properly mix computer science with training planification (Me & Unold, 2011), to personalize training content for individuals and groups.

Recent progress in machine learning and software development has led to the extension of intelligent tutoring systems into non-cognitive skill domains and has resulted in the development of new design architectures and tutoring strategies.

This experiment presents the development of a tutoring model that includes a contextual multi-armed bandit algorithm for the online creation of teaching sequences. The tutoring model is part of a psychomotor ITS for anatomical adaptation training, which is the first phase of sports training and aims to balance the body and prepare it for specialized training (such as football training).

Reinforcement learning was selected for personalization to address various challenges, including limited training time, the complexity of user characteristics, and motivation management.

Most theories and practices in the psychomotor development field are based on research conducted on men. However, it is believed that including women in research may lead to interference due to their physiological variability (such as the menstrual cycle and the effects of oral contraceptives). Therefore, personalization of motor skills learning must take into account significant differences in men's and women's physiology.

The experiment addresses female-specific issues in the development of psychomotor skills, such as strength levels, the menstrual cycle, and the female athlete triad. The principles and parameters of individualization based on sexual dimorphism are then presented.

The proposed tutoring system was initially tested in a virtual environment without taking the dimension of sexual dimorphism into account. In these systematic experiments, groups of trainees followed various personalization strategies.

Subsequently, a simulation involving hundreds of virtual men and women was conducted, which showed that taking sexual dimorphism into account in the tutoring module is essential for injury prevention and for providing personalized sessions for women's training.

8.3 Accounting for Sexual Dimorphism in Psychomotor Intelligent Tutoring Systems

8.3.1 Basis of Human Sexual Dimorphism

The morphological, cognitive, and physiological differences between males and females impact the content development of the training sessions, together with associated risks of injuries and psychological disorders. This section briefly describes these differences.

The difference between women's and men's skeleton size and body composition (i.e., measurements of density, ratio of fat to lean tissue, and muscular body mass adjusted by height) varies at different age periods (Kirchengast, 2010) (Joyce & Lewindon, 2016) (Shephard, 2000). These differences increase at puberty due to hormonal differentiation. On average, adult human males are 7% taller than females, and there is a significantly higher proportion of body fat compared to lean body mass among women. Women have a smaller thorax, a larger abdomen, a broader and shallower pelvis, shorter legs, and a lower relative center of gravity than the male. The distribution of muscle mass in the lower body is relatively similar between the sexes, while females tend to have less muscle mass in the upper body compared to males.

Human performance mainly relies on the anaerobic energy sector (effort duration from 1s to 100s) and the aerobic sector (effort duration from 100s to several hours). The anaerobic sector depends on local stores of phosphagen. Phosphagen store per unit volume of muscle is independent of sex, but as women are less muscular than men, they are disadvantaged.

Females exhibit variations in their absolute strength levels: they generally have from 40% up to 60% of males' strength levels in the upper body and 70% to 85% in the lower body. The rate of force development (RFD) is typically lower in females compared to males. In general, females generate less power than men, and have less power per unit of muscle volume (Shephard, 2000).

Various theories (Baron-Cohen, Knickmeyer, & Belmonte, 2005) (Liutsko, Muiños, Tous Ral, & Contreras, 2020) (Li, 2014) enunciate the difference between women's and men's cognitive abilities

impacting psychomotor skills. In general, women tend to adapt their behavior to their perception of another person's emotions and thoughts. For navigation, they favor an egocentric strategy while using street names and building shapes as landmarks. They outperform males in precision and fine hand abilities, object location and verbal memory, verbal recognition, and semantic fluency tasks.

In contrast, men tend to analyze and explore rules that govern a system. In general, they perform better on mental rotation and spatial navigation tasks than women. For navigation, they tend to favor an allocentric strategy that considers accurate judgments of distance. Men integrate speed and precision more quickly than women, and they tend to be better at sensorimotor tasks, including aiming, catching, and throwing.

The menstrual cycle consists of the physiological phenomena which prepare the woman's body for possible fertilization. Menstruation occurs approximately every 24–35 days. The menstrual cycle comprises three phases: menses (or menstruation), follicular, and luteal. The process contains varying amounts of two endogenous hormones: estrogen (low during the menses, high during the follicular phase, and moderates during the luteal phase) and progesterone (low during menses and follicular phase, increased during the luteal phase) (Chidi-Ogbolu & Baar, 2019). Hormone variations impact aerobic and anaerobic performances (Joyce & Lewindon, 2016), while negatively affecting general outcome of training, bone mass, , chances of attaining optimal body weight, and risk of injury (Pitchers & K., 2019).

Women's morphological differences and hormonal variability during the menstrual cycle induce a higher risk of injury and, in particular, Anterior Cruciate Ligament (ACL) injury. The prevalence of ACL injury for women is 2-10 times greater than in males, for the same psychomotor activities. This prevalence causes a lower rate of force development, hamstring activation deficits, and increased ankle dorsiflexion, along with a valgus position of the knees and external rotation of the hip (Joyce & Lewindon, 2016) (Somerson, Isby, Hagen, Kweon, & Gee, 2019).

Moreover, three interdependent conditions predispose, when combined, female athletes in particular to an increased susceptibility to illness and injury:

- a) "disordered eating (including anorexia nervosa and bulimia nervosa) ";
- b) "osteoporosis (loss of bone density) ";
- c) "amenorrhea (lack of menstrual cycle) ".

8.3.2 Sexual Dimorphism for Scheduling Training Sessions

Traditionally, the schedule and the design of psychomotor skills training sessions rely on physiological temporal variables. The delay between two sessions must be long enough not to induce overfatigue and not too long not to induce detraining. The evolution of session load, which is computed using Equation 1, follows temporal patterns with progressive development from one week to another.

Equation 1. Session Load Formula

$$\text{SessionLoad} = \sum \text{NoRepsEx } k * \text{IntensityEx } k * \text{RestTimeEx } k * \text{NoJointsEx } k$$

The easiest week is set every three weeks to facilitate learning assimilation – for example, one week with easy sessions, one week with medium sessions, one week with heavy sessions, followed again by easy sessions.

Trainee’s performance monitoring and feedback support load adjustment. When scheduling learning sessions for women, tutors must follow the same physiological temporal variables. However, when designing sessions, they must synchronize with the menstrual cycle to define session content. Table 19 describes the relationship between session load and the various phases of the menstrual cycle according to Pitchers (Pitchers & K., 2019).

Table 19. Training load adaptation to menstrual phase adapted from Pitchers

Menstrual Phase	Early follicular	Mid follicular	Late follicular	Early luteal	Mid luteal	Late luteal
Training Load	Light	Medium	Medium / Heavy	Very Heavy	Medium	Light

Trainee profiling before starting and during a psychomotor skill development program generally consists of assessing physical capacities and identifying areas of vulnerability or discomfort related to performance (Joyce & Lewindon, 2016). The consideration of female-specific risks of injuries, particularly ACL and female athlete triad, requires integrating appropriate tests of assessing the injury susceptibility. Results from these tests are used afterward to provide dedicated prophylaxis sessions and adjust learning sessions accordingly.

8.4 Optimizing Teaching Sequences using Machine Learning. Right Exercise at the Right Time (RiERiT) Method

The tutoring process for a Psychomotor ITS was structured on a four-level maturity scale, which includes novice, intermediate, advanced, and expert trainers. Additional temporalities are considered in the adaptation process when moving from one level to another.

The Novice Trainer implements the Multi-Armed bandits' algorithm for personalizing the training sequences in a session. The Intermediate Trainer can personalize a session content; the Advanced Trainer personalizes the micro-cycle, while the Expert trainer can create customized mesocycle and macro-cycle content for each individual.

The underlying model relies on templates of training sequences for generating micro-cycles and sessions based on trainee input, including the number of sessions to train in the current week (associated micro-cycle), time to train for a session, and micro-cycle goals or focus. A sub-list of templates used for generating micro-cycles in anatomical adaptation is presented in Table 20.

Table 20. Micro-cycle Templates Examples for Anatomical Adaptation

Micro-cycle Template Name	# of pieces of Training	Recommended Trainee
Push / Pull / Lower / Upper / Lower	5	Men
Hip Dominant / Knee Dominant / Upper / Lower / Upper	5	Women
Upper / Lower / Full / Full / Full	5	Mixed

Sessions are focused on training fundamental movements or body areas, for example: push (vertical and horizontal), pull (vertical or horizontal), lower body, upper body, full body, hip dominant, and knee dominant. Sessions have specific templates associated with each fundamental movement or body area, as well as trainee availability (30, 45, or 60 minutes). There are also warm-up templates created by sports experts mapped on session-targeted areas. A session template is composed of a list of generic exercises, each of them with a targeted area, charge level, and rest time. *Selfit* recommends exercises that are most likely to improve the overall competence level across all psychomotor components, based on previous trainee performance.

The charge level is represented by the number of repetitions (ranging from 8 to 15) and sets (ranging from 1 to 5). Targeted areas for exercises are represented by either fundamental or complementary movements, and the domain module maps real exercises with movement types and difficulty levels (from 1 to 5).

Trainee level for each fundamental movement is estimated through calibration challenges, which were previously introduced. As such, the *Novice trainer's* challenge is to choose the right exercise from the list of available exercises (called the Right Exercise at the Right Time - *RiEaRiT*).

An efficient online method, namely contextual multi-armed bandits (Lu & Pal, 2010) was used to explore and optimize different exercises and estimate trainee progress. Such algorithms model a situation where a decision is taken in a sequence of independent trials based on a given context, which contains side information.

The context of *Selfit* is represented by the trainee shape-of-the-day, which is computed using a Borg scale (Spielholz, 2006), i.e., a CR-10 (Category Ration-10) scale to measure different body shape parameters. The selected algorithms ensure the creation of a personalized learning experience relying only on limited domain knowledge. The goal of the model is to maximize the total pay-off, or reward, of the chosen actions.

The reward of the Multi-Armed Bandit, after choosing an exercise, is computed as the difference between external load (EL) – considered exercise charge: number of repetitions and number of sets and internal load (IL) – computed from estimated user shape, subjective value. This difference is also defined as the number of Repetitions in Reserve (RiR) (Hackett, Johnson, Halaki, & Chow, 2012) and, for anatomical adaptation training, the best values are positive, as close as possible to 0.

RiR denotes how many more repetitions a trainee could have performed at the end of a set. 0 indicates that the number of repetitions provided is the maximum number of repetitions that the user could have performed. A positive value reflects the number of potential repetitions that could have been performed; nevertheless, this value is subjective. A negative value means the trainee has failed at that set; if, for example, the set had 12 reps and RiR was -2, the trainee was able to perform only 10 reps. While following a training program, there are specific sessions that require the trainee to reach failure (negative RiR).

The reward is computed based on the formula in Equation 2. Valid values of RiR are integer values in [-10,10] interval. If the reported RiR is either 0 or 1, the reward is 1, the highest value. This means the user was able to execute the number of repetitions of that exercise, and it was highly challenging also. If the reported RiR is greater than 1, the reward is positive, in the [0.1, 0.5] interval. The higher the RiR value is, the smaller the reward.

Equation 2. Contextual Multi-Armed Bandit' Reward Formula

$$rew(RiR) = \begin{cases} 1 & , \text{where } RiR \in \{0,1\} \\ \frac{1}{RiR} & , \text{where } RiR > 1 \\ \frac{1}{|RiR|} - 1 & , \text{where } RiR < 0 \end{cases}$$

If the reported RiR is less than 0, this means the user has failed the current exercise. The reward is proportional to the RiR variability. The higher the RiR value is, the higher the reward. For this branch, the reward is negative, with values within the $[-0.9, 0]$ interval.

9 *Selfit* – A Psychomotor Tutoring System prototype in Open Environments

9.1 Résumé

Selfit est un prototype de système de tutorat intelligent (STI) pour le développement psychomoteur qui vise à tester le cadre en situation réelle. Il se compose de quatre éléments en interaction : L'interface utilisateur graphique (y compris l'authentification, l'étalonnage, le dialogue et la session de formation), le modèle du domaine, le modèle de l'étudiant (y compris le suivi) et le modèle de tutorat.

Le modèle de domaine Selfit soutient le processus d'apprentissage en fournissant des informations sur les objectifs d'apprentissage, l'évaluation du stagiaire, la définition du programme d'apprentissage et l'adaptation. La connaissance du domaine est structurée à l'aide de l'ontologie OntoStrength, qui décrit les relations entre les chaînes musculaires, les mouvements articulaires et les muscles pour le développement de la force, ainsi que les différentes modalités des programmes d'entraînement. Le modèle de domaine comprend plus de 1 000 exercices provenant de diverses sources et est décrit en termes de nom d'exercice, de famille de mouvements, de niveau, de mode d'exécution, de zone de mouvement, de dynamique des mouvements, de nombre d'articulations impliquées, de type de corps, etc.

Le modèle Selfit Student stocke des informations sur les stagiaires, telles que des informations générales, l'état mécanique, les problèmes physiologiques, les exercices préférés, etc. Le modèle de l'élève est constamment mis à jour pendant le programme de formation et utilisé pour générer des sessions de formation et suivre les progrès. Le modèle de tutorat Selfit soutient le processus d'apprentissage en adaptant le programme d'entraînement aux caractéristiques du stagiaire à l'aide d'un algorithme de bandit à bras multiples. L'algorithme ajuste le contenu de la séance d'entraînement en fonction d'éléments tels que l'état de préparation à la formation et l'évaluation subjective des effets de la tâche fournie par l'utilisateur. Le modèle prend en compte la dimension du dimorphisme sexuel et met en œuvre la méthode Right Exercise at the Right Time (RiERiT) pour l'adaptation dans le domaine psychomoteur.

Le système Selfit utilise une application Web progressive (PWA) comme interface utilisateur graphique. Il permet aux utilisateurs de s'enregistrer avec un nom d'utilisateur, un mot de passe et une question de sécurité, puis d'accéder au programme de formation en remplissant d'abord un défi

d'étalonnage. Le module d'interface est composé de l'élément d'authentification, de l'élément d'étalonnage/évaluation, de l'élément de retour d'information et de l'élément de session de formation, qui sont décrits dans le présent chapitre. Ensuite, une mise à jour de l'architecture existante, qui inclut la composante du dimorphisme sexuel, est présentée.

Le moteur de génération d'exercices Selffit génère des séances d'entraînement personnalisées en fonction de l'historique d'entraînement de l'utilisateur, de ses niveaux de fatigue et de ses préférences en matière de développement musculaire. Le système met également à jour son ontologie avec les commentaires de l'utilisateur après chaque exercice. Le programme d'entraînement s'ajuste jusqu'à ce que l'objectif d'entraînement de l'utilisateur soit atteint, et des défis d'étalonnage sont périodiquement réalisés pour mettre à jour les niveaux de compétence de l'utilisateur. Enfin, ce chapitre présente un module d'évaluation psychomotrice utilisant la vision par ordinateur pour évaluer le risque de blessure au LCA. Ce module capture des vidéos de l'utilisateur effectuant un squat arrière et utilise une bibliothèque de reconnaissance des mouvements humains pour identifier les points clés du mouvement de l'utilisateur. Le module d'évaluation des risques évalue ensuite la distance entre le genou et la cheville de l'utilisateur pour déterminer s'il y a un valgus, ce qui augmente le risque de lésion du LCA.

9.2 Introduction

To test in real life our framework, in an open environment, and for the mass population, we developed *Selffit* - see Figure 38, a prototype of ITS for psychomotor development with special care for user ITS interface modeling.

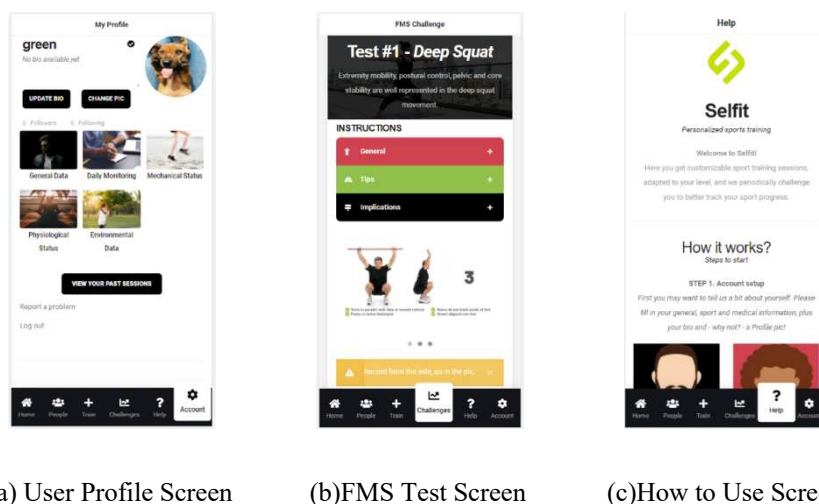


Figure 38. Selffit Graphical User Interface

The chapter is structured as follows. Next, the *Selffit* architecture overview is presented, where the main ITS components in the proposed system are described – Domain, Student, Tutoring, and

Interface. Then, the sexual dimorphism dimension is considered, and design updates are presented. The first *Selfit* implementation did not consider sexual differences, and it was shown in II. 0 to be of high interest in psychomotor development. Next, workout generation rules and exemplification of the user experience while training are presented. Finally, an automated movement detection experiment using computer vision is conducted to outline the importance of movement execution. This is highly important for preventing the risk of injuries, based on sexual differences.

9.3 *Selfit* Architecture

Four conceptual components, or modules, interact with each other: a) Graphical User Interface (including Authentication, Calibration, Dialogue, and Training session subcomponents), b) Domain model, c) Student model (including Monitoring), and d) Tutoring model.

9.3.1 Domain Module

The *Selfit Domain model* assists in the learning process by providing responses to a) requests related to learning objectives definition, b) trainee evaluation, c) learning program definition, and d) adaptation by answering requests. Declarative (movement skills, physical capacities, and exercises), procedural (training sessions generation and feedback interpretation algorithms), and conditional knowledge (physical status) map answers to requests.

The OntoStrength ontology underlies the *Selfit* domain model, and it consists of a "movement skill class with associated psychomotor profile, movement patterns, and training program modalities". It also outlines the relationships between "muscle chains, joint movements, agonist, antagonist, and synergist muscles" in the training of strength qualities. It also outlines various development modalities with associated load patterns. More details on the domain modeling are presented in II. 7.3.3 – OntoStrength sub-domains description.

The domain model also indicates the knowledge of the sports training exercises. More than 1.000 exercises have been gathered from multiple sources or created by our team of sports scientists. The exercises were described based on the following parameters:

- Exercise name;
- Movement family: basic or complementary;
- Level, on a scale from 1 to 12;
- Execution mode: time-based or repetition-based;
- List of materials;

- Movement area: one of the six areas which are assessed in the calibration challenges (push vertical/horizontal, pull vertical/horizontal, hip dominant, knee dominant);
- Dynamics of movements – a description with the list of movements implied (e.g., elbow flexion, shoulder girdle downward rotation, shoulder extension, spine rotation);
- Number of joints involved;
- Body type – unilateral, or bilateral;
- Explanation - a description to clarify how the executions should be performed, written both in English and French; an example is "start with your weakest leg", or "if you have long forearms, do not go below the ears when performing the movement";
- Video – which was either recorded by us, more than 200 exercises, or from the internet.

The *Selfit Domain* encapsulates the logic for the calibration session, used to assess the trainee’s level. 24 exercises were described, 4 for each movement area, with incremental difficulty and a protocol for execution. The domain also has a complete mapping of the movements with the body muscles. The trainee has the option to configure a session by specifying the desired muscles to train. The list of muscles selected is further mapped to the corresponding movements and based on trainee level and current session goal, the desired exercises are proposed.

An example of mapping of two specific body areas – hip and knee, with the decomposition in possible movements and the corresponding muscles can be seen in Table 21. Each body area can be split into one or more possible movements. Each movement can involve one (e.g., knee extension) to nine (e.g., hip lateral rotation) muscles.

Table 21. *Selfit Domain model*. Hip and Knee mapping of body area – movement – muscles involved.

Body Area	Decomposition of Movements	List of Muscles Involved ¹⁵
Hip	"Flexion"	"Iliopsoas; Tensor Fasciae Latae; Rectus Femoris; Sartorius; Adductor Longus; Adductor Brevis; Pectineus"

¹⁵ Ryan Hoyme, “Muscle Actions Explained - Massage, Massage Videos, Massage Pictures, Massage Tests and More”, link: http://www.massagenerd.com/Muscle_Actions_Explained.php, (Accessed on 7.01.2023)

	"Extension"	"Gluteus Maximus; Semitendinosus; Semimembranosus; Biceps Femoris (long head); Adductor Magnus (ischial fibers) "
	"Adduction"	"Adductors; Pectineus; Gracilis; Gluteus Maximus (lower fibers) "
	"Abduction"	"Gluteus Medius; Gluteus Minimus; Tensor Fasciae Latae; Sartorius"
	"Transverse Adduction"	"Adductors; Pectineus; Gracilis; Tensor Fasciae Latae"
	"Transverse Abduction"	"Gluteus Maximus; Gluteus Medius; Gluteus Minimus; Piriformis; Obturator Externus"
	"Medial Rotation (Internal) "	"Tensor Fasciae Latae; Gluteus Medius; Gluteus Minimus (anterior fibers) "
	"Lateral Rotation (External) "	"Gemellus Superior; Gemellus Inferior; Obturator Internus; Obturator Externus; Quadratus Femoris; Piriformis; Piriformis; Gluteus maximus; Sartorius; Gluteus Medius"
Knee	"Flexion"	"Hamstrings; Gracilis; Sartorius; Popliteus; Gastrocnemius"
	"Extension"	"Quadriceps Femoris; Tensor Fasciae Latae"
	"Medial Rotation (Internal) "	"Popliteus; Semimembranosus; Semitendinosus; Sartorius; Gracilis"
	"Lateral Rotation (External) "	"Biceps Femoris"

The muscles involved are shared between the movements, it can be seen that for example Piriformis is involved in three of the movements presented in the table: Hip Transverse Adduction, Hip Medial Rotation, and Hip Lateral Rotation. 10 body areas were described in our modeling: shoulder, neck,

shoulder girdle, elbow, hip, knee, ankle, spine, radioulnar, and wrist and midcarpal, each with two to ten movements decomposed. Elbow was split into two possible movements, flexion, and extension, while the Shoulder has been split into ten decomposed movements.

The session templates used in the training program include generic exercises, targeting specific movement areas, or complementary. The movement areas are the ones defined and used also in calibration and include six categories – push horizontal and vertical, pull horizontal and vertical, hip dominant, and knee dominant. A session template for a pull session of 30 minutes, for example, includes the following generic exercises: pull horizontal exercise 1, pull horizontal exercise 2, pull vertical exercise, and two complementary exercises.

The trainee can configure the desired muscles to train which are linked to the desired movements. Another mapping has been performed by our sports scientist between the generic session exercises – movement area and movements, and this can be seen in Table 22. This mapping allows the tutor to select from a filtered list of exercises, which contain one or more muscles pre-selected by the user.

Table 22. *Selfit Domain model*. Mapping of Movement Areas with Body Areas Movements

Movement Area	Body Area Movements
Push Horizontal	Shoulder transverse flexion; Shoulder flexion; Shoulder girdle protraction; Elbow extension
Push Vertical	Shoulder flexion; Shoulder abduction; Shoulder girdle upward rotation; Elbow extension
Pull Horizontal	Shoulder extension; Shoulder girdle retraction; Elbow flexion
Pull Vertical	Shoulder adduction; Shoulder girdle downward rotation; Elbow flexion
Hip Dominant	Hip extension; Knee extension; Spine extension; Ankle plantar flexion
Knee Dominant	Hip extension; Knee extension; Ankle plantar flexion

Complementary	Knee flexion; Hip adduction; Hip abduction; Hip Flexion; Hip internal rotation; Hip external rotation; Ankle dorsiflexion; Shoulder internal rotation; Shoulder external rotation
---------------	---

The modeling of the muscles, including the hierarchy of muscles, movements, joints, and dynamics of movements involved in each exercise are at the core of domain modeling in the *Selfit* system. This module continuously exchanges information with the tutor module to provide tailored sessions to the trainees.

9.3.2 Student Module

The *Selfit Student Model* includes information about trainees' psychomotor abilities, particularly those related to the super-compensation cycle status. Moreover, it includes usage statistics. The *Monitoring* module accesses information about how students are using the system and how they progress with their training. The module uses this information to modify the training parameters. The *Selfit Student Model* supports the generation of training sessions and the monitoring of trainee efficiency to optimize progression while ensuring motivation to practice and progress. More details on the student modeling imported from *OntoStrength* are presented in 7.3.3.

The student module maps trainee characteristics and is updated constantly during the training program. General information about the trainee is stored, and this includes the nickname, gender, birth year, profession, height, weight, and bio. Only the nickname field is required, the others are optional. The more information user feeds within the *Selfit* system, the better the recommendations will be. For example, if the trainee will not input the gender, by default the training program will be man specific. *Selfit Student Module* also maps information about the trainee's mechanical status. This includes specific events reported by the user – pain, injury, surgery, or others, on one or more body regions: back, torso, upper extremity, lower extremity, and head and neck. This data, which can be updated at any type by the trainee, are reflected in the training planification. The reported events act as restrictions on specific movements and muscles to be used while training.

Selfit Student Module maps a list of physiological issues, with potential medical risks, which are also reported by the trainee. This includes a form the trainee can opt to fill in, which includes the following health risks – hypertension, diabetes, post-cancer, and obesity. The physiological status set a list of restrictions on the generation of the training program. Other data specific to the user is a list of favorite exercises, which are marked by the user while training and will increase the chances of being recommended further. Each user has a list of other trainees they follow, and who follow them in the *Selfit* application.

Also, the user periodically reports his physical shape, and this is stored in his profile – this includes the fatigue level, motivation to train, sleep quality, and stress level. The history of each session configuration is stored at the user level, which includes – the location to train (home or at the gym), available time (between 30 to 90 minutes), available materials (specific to the training location, the user has the option to remember them), preferred muscles to train. The *Selfit Student Module* stores the history of the reported Repetitions in Reserve for each exercise proposed to the user. The cumulated reward per exercise is also stored at the user’s level and is used for tailoring future sessions.

9.3.3 Tutoring Module

The *Selfit Tutoring model* helps the learning process by offering machine learning mechanisms to adapt the learning program to the trainee's characteristics. The tutoring module required integration with the psychomotor development domain. Sport training is a complex process, which supports adaptation and personalization while considering different temporalities (e.g., exercise, session, week, month). This domain is related to the temporal organization of the development program by considering the objectives and content of the different training blocks (macro-cycle, mesocycle, micro-cycle, or workout). The training development program periodization has specific rules for each training block. An example is the Micro-Cycle class definition and relationships, which can be seen in a GraphDB visual representation in Figure 39.

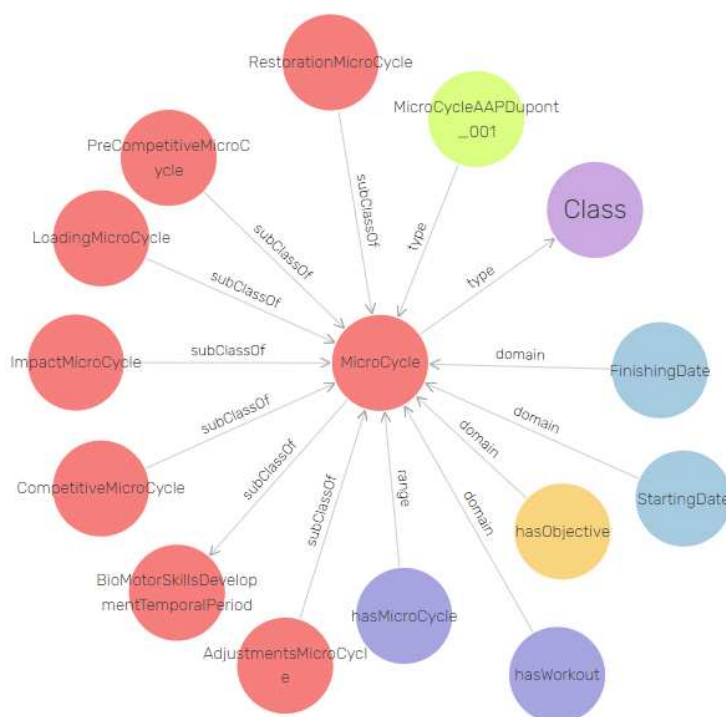


Figure 39. GraphDB View of Micro-Cycle Program Periodization.

Moreover, the psychomotor domain is also related to the specific development rules and constraints of the different sub-skills used by the development program. Thus, *OntoStrength* represents different abilities such as strength, endurance, or speed, all linked to associated development rules.

Selfit currently focuses on the first level of adaptation – the *Novice Trainer*, which supports adaptation at the workout level, based on a list of micro-cycle templates. Examples of templates used to generate a micro-cycle are presented in Table 20 – templates include the number of sessions in the micro-cycle, defined as a week of training (3 or 5 sessions), with a micro-cycle focus, and considering the sexual dimorphism dimension. The current *Selfit* version does not support adaptation for the higher training levels - mesocycle and macrocycle, namely *Intermediate Trainer* and *Advanced Trainer*.

A multi-armed bandit algorithm supports the definition of training workouts by adjusting the template content with inputs related to readiness to train, the effective realization of training tasks, and subjective assessment of task effects provided by the user. The sexual dimorphism dimension is also considered due to morphological, cognitive, and physiological differences between males and females. The design of training sessions relies on physiological temporal variables.

The delay between two sessions must be long enough to not induce overfatigue and not too long not to induce detraining. For men, the evolution of session loads follows temporal patterns with progressive development from one week to another. The easiest week is set every three weeks to facilitate learning assimilation – for example, one week with easy sessions, one week with medium sessions, one week with heavy sessions, followed again by easy sessions. For women, the evolution of session loads follows their menstrual cycles by using a specific template.

Selfit implemented the Right Exercise at the Right Time (*RiERiT*) method, which has proven efficient for adaptation in the psychomotor field, as shown in II.8. The tutor implements the multi-armed bandit algorithm for personalizing the training sequences in a session, based on session templates. A session template has a list of generic exercises, each of them with a targeted area, charge level, and rest time. *Selfit* recommends exercises that are most likely to improve the overall competence level across all psychomotor components, based on previous trainee performance.

9.3.4 Interface Module

The *Selfit Graphical User Interface* supports exchanging information between the trainee and *Selfit* to facilitate the learning process. Students access the interface module through a Progressive Web App,

available for cellphones, tablets, or personal computers. *Selfit* is accessible in the most popular mobile stores – Google Play¹⁶ and App Store¹⁷.

Progressive Web Application (PWA) development is a novel platform that combines the capabilities and experiences of native applications with the reach of the web (Biørn-Hansen, Majchrzak, & Grønli, 2017). PWAs are recognized for a set of key features, such as responsive, connectivity-independent, app-like, safe, or installable. The application acts as the User Interface Module and it is responsible for updating the Student Module, through continuous user feedback.

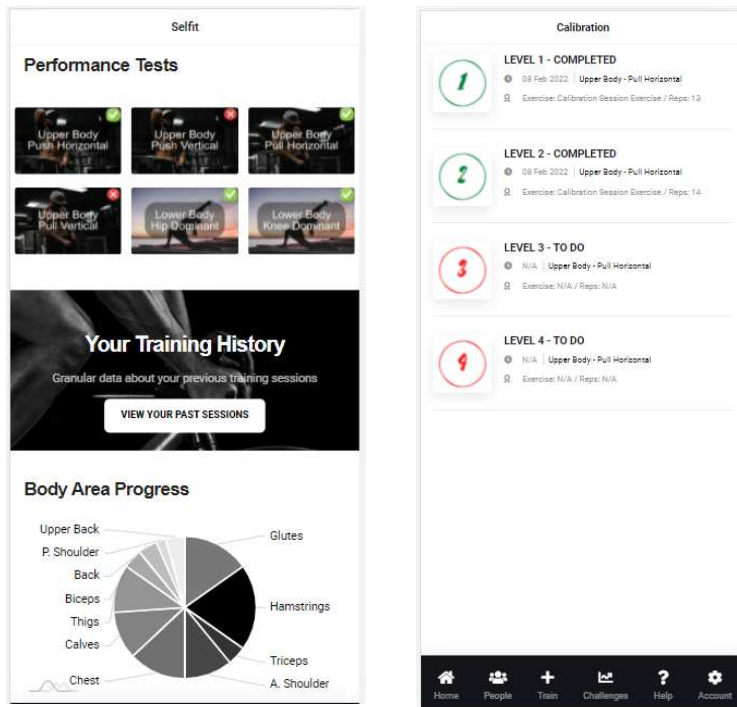
The flow will be described further. First, the user needs to register within the application with a username, a password, and a security question. Afterward, they are automatically logged into the system, and they can either fill in more details on their profile page or start the calibration challenge. This challenge is the first session all the users are required to perform before enrolling in the training program. Users will be redirected to this screen when they start the first training.

The calibration challenge is followed by the training sessions, which are configured based on the trainee level and preferences. The calibration can be performed again at every moment but, to have any observable progress, it is recommended to perform it once every month (or after every 12 sessions). *Selfit* allows users to check their past sessions and view a summary of body areas trained. Also, a user can follow other users' progress and compare training data on the People screen.

The interface module is composed of the authentication component, the calibration or assessment component, the feedback, and the training session. *Selfit* does not track and does not store any personal user data; user profiles consist of a nickname, password, and a security question managed by the *Authentication* subcomponent. The *Calibration* subcomponent supports the definition of each student's learning motivation by selecting physical qualities to develop. Then, it supports the trainee's initial and intermediary evaluations – see Figure 40.

¹⁶ <https://play.google.com>

¹⁷ <https://www.apple.com/app-store>



(a) Trainee Summary

(b) Upper Body Calibration Summary

Figure 40. *Selfit* Interface –Calibration subcomponent

For each physical quality, the component provides a testing protocol. For example, the evaluation for strength qualities development consists of a challenge aiming to perform four of the six essential movements categories (upper body push horizontal, upper body push vertical, upper body pull horizontal, upper body pull vertical, lower body hip dominant, lower body knee dominant), one exercise for every four levels of difficulty. Calibration sessions should be performed regularly to adjust the user’s progress while training.

Feedback is crucial to performing motor skills well (Bilodeau & Bilodeau, 1961). The *Dialogue* subcomponent supports students in providing information before and after training sessions to help adapt training sessions to students’ shape and availability. Before starting a session, *Selfit* asks students to self-evaluate their fatigue level, motivation to train, sleep quality, and stress level on a scale from one to ten.

During training sessions, *Selfit* asks students to self-evaluate at the end of each content exercise and answer whether they could perform additional repetitions and, if yes, how many. After the training session, *Selfit* asks the users to input the session difficulty they perceived on a scale from one (very hard) to ten (very easy).

The *Training session* subcomponent provides learners with a training session description, which includes a summary of warmup, content, and cooldown exercises. Training sessions are easily

configurable; trainees can select train location (home or gym), with many session templates available, the training materials available (barbell, elastic band, machine, etc.), and complementary muscles to target while training (anterior shoulder, biceps, forearms, thighs, etc.). While training, a video demonstrating the movement and details required to accomplish the correct load (i.e., number of sets, number of repetitions, rest between repetitions and between sets) is displayed for each exercise.

In the recent literature review of authoring tools in ITS models (Dermeval et al., 2018), it was shown that the "Display learners' statistics" feature was the most common -12.12%, by far, in the ITSs built in the last 15 years. For this reason, learners' statistics are also shown in *Selfit*, an example can be seen in Figure 40 (a) – body area progress. Learners can see their full training history, per microcycle and session, and also the progress they did per trained body area, as in Figure 41.

While training, a video demonstrating the movement and details required to accomplish the correct load (i.e., number of sets, number of repetitions, rest between repetitions and between sets) is displayed for each exercise. Features to improve user experience have been implemented in the graphical user interface, such as auto mode, fast mode, or resume a session. Auto mode feature is helping learners to move automatically between time-based exercises, without user input – manually pressing the "Next" button. The fast mode feature enables the user to significantly decrease the time of a session, which will group exercises two by two, using the superset training concept – both exercises are executed together, as one set. This feature is useful when the learner is running out of time while training. *Selfit* Training session subcomponent can be seen in Figure 41 (a), (b), and (c).

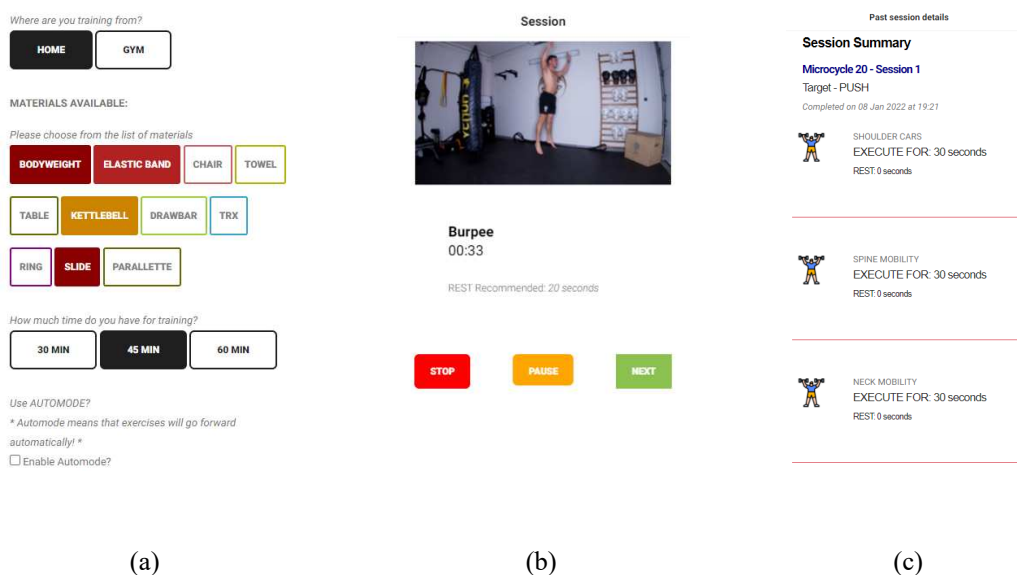


Figure 41. *Selfit* Interface – Training Session subcomponent

Users can restart an ongoing session, in the case of an internet connection failure, or if he/she mistakenly closes *Selfit*. The system tracks user usage behavior, such as the time spent per exercise, details about the screens accessed, or overall time spent in the application.

9.3.5 Sexual Dimorphism Integration. *Selfit Version 2 Architecture*

An updated version – *Selfit Version 2*, has been further developed, which now considers the gender dimension. The updated system includes a Monitoring module, required for the assessment of risk injury while training, as well as a novel tutoring approach based on a menstrual cycle-specific mesocycle calendar.

Updates were also performed on core ITS components, namely the Domain, Student, and Tutoring models. The updated version of the architecture can be seen in Figure 42, with the new sub-modules highlighted in blue.

A Updates on the Domain module

OntoStrength assists in the learning process by offering responses to requests related to a) learning objectives definition, b) trainee evaluation, c) learning program definition, and d) adaptation to student characteristics. The ontology core consists of the following classes: movement skill, the profile of trainee psychomotor skills, motion characteristics, and modalities for a training schedule. It was updated to provide answers to requests related to: "Which session content is the most appropriate to the student, based on her position in her menstrual cycle? ", "What is the student risk status? ", and "How the learning session has to be updated? ".

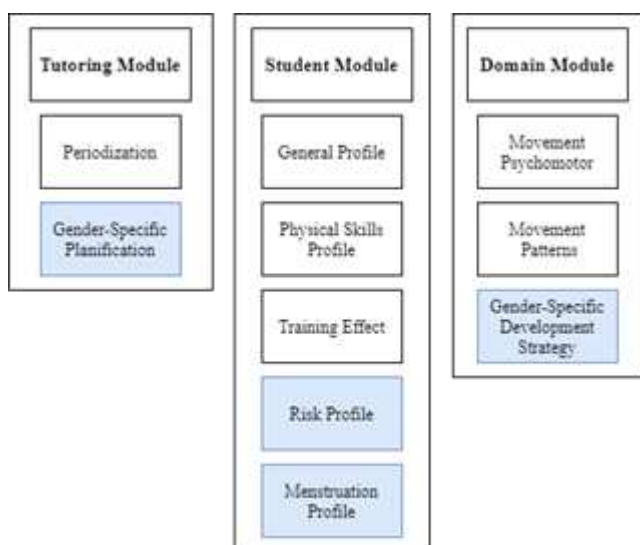


Figure 42. *Selfit* Updated Architecture considering Sexual Dimorphism

Consequently, classes related to requirements for designing sessions based on different phases of the menstrual cycle and classes describing female-specific risks of injuries and associated tests and prophylaxis measures have been added to the initial ontology.

B Updates on the Student module

The *Selfit* Student model includes information on the trainee's psychomotor skills abilities, particularly those related to the super-compensation cycle status. The Monitoring module gathers data on how the trainees are utilizing the system, along with their daily progress, and accordingly adjusts the training parameters.

The model supports the generation of training sessions and the monitoring of trainee efficiency to optimize progression while ensuring motivation to practice. The sexual dimorphism update consists of the addition of two dimensions in the student model. The first one aims to describe the evolution of female-specific risks while training and to introduce the appropriate mitigation tasks to be performed when a risk value exceeds an imposed threshold. The second dimension aims to integrate the evolution of women's menstrual cycle and to properly select training session loads based on the women's calendar.

C Updates on the Tutoring module

The *Selfit* Tutoring model supports the learning process by providing training rules at different training levels (session, micro-cycle, mesocycle, and macro-cycle) and machine learning mechanisms to support the learning program's adaptation to the trainee's characteristics.

Sport training is a complex process, which supports adaptation and personalization while considering different temporalities: exercise, session (30 min to 120 min), micro-cycle (one week), mesocycle (one month), and macro-cycle (three months). Starting with pre-defined templates for micro-cycles and sessions, a Novice Trainer generates the most appropriate learning task using a Multi-Armed Bandit strategy (Clement et al., 2015)

The task consists of exercise, level, number of repetitions, and number of sets, by considering student characteristics, history, and his current state. The trainee provides direct feedback in the system, specifying the number of Repetitions in Reserve (defined as the number of repetitions he could have performed until failure), Rate of Perceived Effort, and Fatigue Perception.

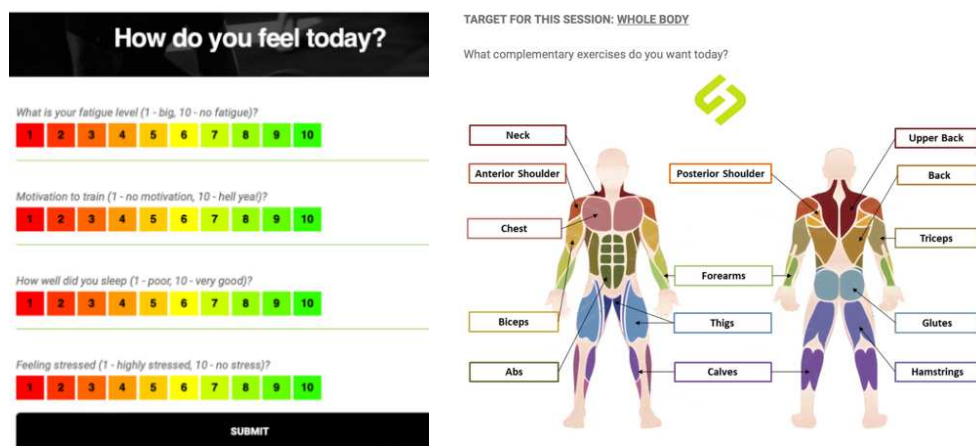
The ratio between the planned and the perceived effort supports the evaluation of the session effectiveness and the baseline for algorithm learning. The dynamic of charge (i.e., complexity of exercises, number of repetitions, number of sets, and load) follows a calendar with a set of training

rules defined by sports coaches. The charge is cataloged as easy in the first week, then medium, heavy, followed again by easy. The sexual dimorphism update consists of a calendar differentiation considering the dynamic of charge for men versus women and integrating the menstrual cycle. The training session content and charge are defined based on the trainee’s menstrual phase.

9.4 Workout Generation and Monitoring the Training Impact

The *Selfit* workout generation engine defines the content of the next workout session by using information about students’ characteristics, past performances, and current fatigue levels. The tutoring module first identifies the workout target by predicting the most accurate template, based on the student’s training history and actual fatigue signature. Once selected, the module generates appropriate content using training strategies and students’ personalization signatures. The user can also customize training content by modifying his/her preferences about developing muscles, before starting a workout. At the end of each exercise, session phase, and workout, *Selfit* assesses success or failure and asks students about their perceptions of the effort.

The *Selfit* user interface module updates the ontology with training information updates submitted by users, such as the daily fatigue profile, or the perception of effort after achieving a training task. An example of the graphical interface inside the training session can be seen in Figure 43 (a), where the user subjectively assesses his physical shape before starting a training session; muscle development preferences are described in Figure 43 (b).



(a) Daily fatigue monitoring interface (b) Muscle development preferences interface

Figure 43. *Selfit* Interface – Parameters for Workout Generation

The training program is continuously adjusted until the trainee development objective is achieved. Calibration challenges are required to be performed periodically for updating student competency levels and to provide personalized training sequences. The parameters used for training personalization are:

- user levels on each movement area computed on calibrations;
- user reported state, before each session – set of parameters which assess current state, including fatigue level, stress level, or motivation to train;
- training session configuration – available materials, preferred muscles to train.

User feedback after each exercise is stored and the tutoring module improves recommendations based on the reported repetitions in reserve. Personalization is perceived at the session level. Before starting each training, the user configures the current session parameters' and then all the exercises are generated. Afterward, the session summary is shown to the user and then the training can be started.

9.5 Psychomotor Assessment using Computer Vision. A Study on Mitigating the Risk of Injuries

A women's ACL injury risk assessment module was developed to support student initial screening and injuries risks monitoring. The assessment process is structured into four phases, see Figure 44. The first phase consists of capturing one frontal and one sagittal video of the student performing a back squat. In the second phase, a human motion recognition module provides a discretization of each body joint's trajectory while performing the back squat. Then, the risk assessment module analyzes this model for calculating the ACL injury risk factor. Finally, if risks are detected, the module provides instructions to the ITS.

Phase 1. The first phase involves recording a front and side video of the student performing a back squat. The back squat exercise helps identify biomechanical deficits, such as inefficient motor unit coordination or recruitment, muscle weakness, strength asymmetry or joint instability, joint immobility, or muscle tightness (Pitchers & K., 2019).

The two videos allow having a complete perspective of the performed movement and then provide information to assess knee valgus and ankle flexibility, two of the essential factors involved in the ACL injury risk.

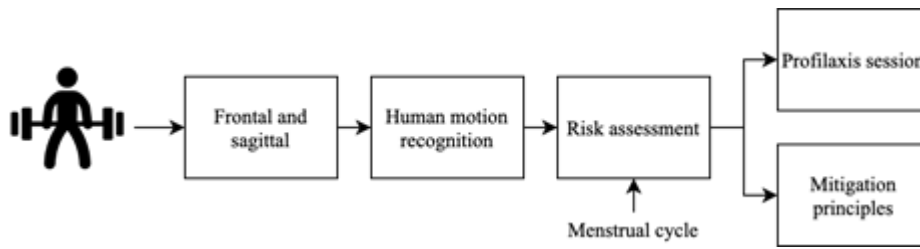
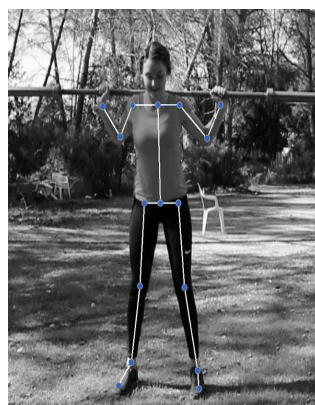


Figure 44. ACL injury risk assessment module process

The two videos are necessary to have a complete perspective of the performed movement. For example, when analyzing the knee joint movement, the sagittal view allows calculating the angle between the student's hip, knee, and ankle, and the frontal view gives information about the distance between the student's knee and the vertical line passing through the middle of its ankle. The combination of these values supports the assessment of the knee valgus.

Phase 2. In the second phase, a human motion recognition module provides a discretization of each body joints' trajectory performing the back squat. The OpenPose 2D pose estimation library (Cao, Simon, Wei, & Sheikh, 2018; Simon, Joo, Matthews, & Sheikh, 2017) allows obtaining videos with an added overlay containing the body key points and lines and a collection of JSON files, one for each frame, containing the position of the essential points, in pixels, as it can be seen in Figure 45. A recent study (Ota et al., 2020) demonstrates the reliability and the validity of this library results by comparing them to results provided by a kinematic measurement by three-dimension motion analysis devices using VICON.



- "Neck"
- "LShoulder"
- "LElbow"
- "MidHip"
- "LHip"
- "LKnee"
- "LAnkle"



"LSmallToe"

"LHeel"

Figure 45. ACL Injury Risk-Assessment Module demonstration

Phase 3. The risk assessment module analyzes this model for calculating the ACL injury risk factor. The monitoring of the inner aspect of either knee passing the inner ankle bone from the front perspective during any phase of the squat helps identify a valgus (Somerson et al., 2019). The module evaluates the distance between "R or L Knee" and "R or L Ankle" when the student performs the back squat.

If this distance is negative and greater than five centimeters, it means that the trainee has a valgus. The following figures describe a case where there is no valgus (see Figure 46) and a case where there is a valgus on the right knee Figure 47 a.

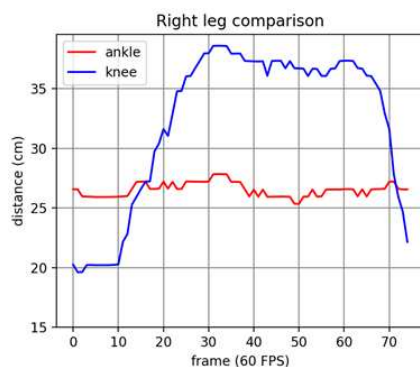


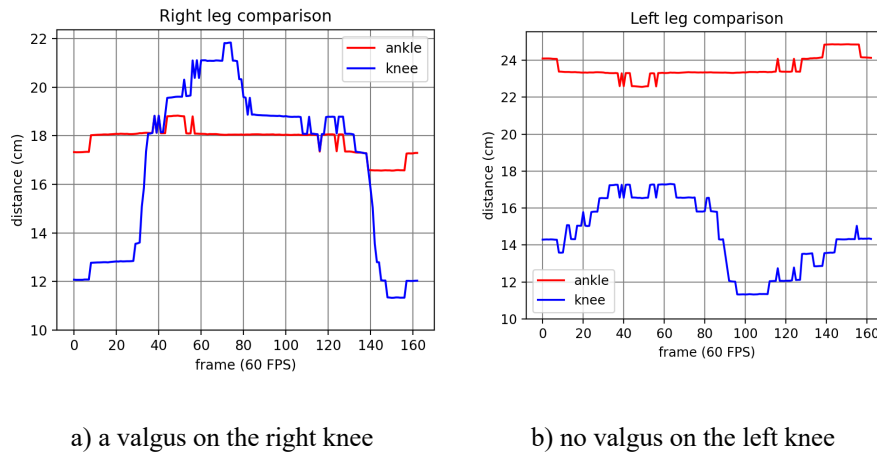
Figure 46. Illustration of the evolution of the distance between student's knee and ankle without valgus.

In the first case, the squat movement starts in frame 10 and finishes in frame 70. At the initial position, the distance between the knee and ankle is negative but less than 5cm. As soon as the movement starts, the distance positively increases and then decreases until the movement is over. This is the behavior of a student without valgus.

In the second case (see Figure 47 a), the squat movement starts in frame 30 and finishes in frame 150. At the initial position, the distance between the knees and ankles is negative, and the distance between knees and ankles is superior to 5 cm, demonstrating the presence of a valgus position before the start of the movement. During the movement, the negative distance between the left knee and the left ankle increases, demonstrating the valgus position's increase.

On the left side, the distance decreases and becomes positive during the movement showing a correct position. At the end of the movement, the distance between knees and ankles demonstrates a valgus position for both sides.

Phase 4. The risk of ACL injury increases if both a valgus and flexible ankles are detected. During the ovulation period of the menstrual cycle, hormones maximize the risk. If the risk assessment module detects a risk of ACL, it communicates with the *Selfit* ITS by providing mitigation rules to redesign the planned and future learning sessions for considering low-risk tasks. Furthermore, *Selfit* provides additional prophylactic learning sessions to decrease the risk of ACL with specific strength, flexibility, and proprioceptive tasks.



a) a valgus on the right knee b) no valgus on the left knee
 Figure 47. Illustration of the evolution of the distance between student's knees and ankles with variations between left-and right

The current chapter introduces a comprehensive description of *Selfit*, an ITS for psychomotor development, alongside encountered design and implementation challenges and architectural considerations for the sexual dimorphism dimension. In terms of future extension, I envision the integration of live recordings for exercise execution with computer vision techniques to perform automatic counts of repetitions, avoid user manual input errors, and to enhance the overall user experience. Also, the next *Selfit* version should include Natural Language Techniques to improve learner-tutor communication. Currently, trainees need to manually input feedback for each set of exercises in the graphical interface; instead, a voice-based interaction module may also improve user experience.

10 Results

10.1 Résumé

Les expériences visant à valider Selfit ont été menées à la fois par simulation et dans une configuration réelle. Les résultats comprennent la répétition en réserve (RIR) par population et les gains d'apprentissage au fil du temps. Les résultats d'une enquête sur l'expérience des utilisateurs ont également été présentés. L'expérience basée sur la simulation visait à déterminer l'efficacité du module de tutorat en simulant différents algorithmes de bandits à bras multiples. L'expérience a été menée dans un environnement virtuel avec 1800 exercices, divisés en 5 niveaux de difficulté, et une population de 1000 stagiaires. Les agents ont été entraînés avec 4 stratégies différentes, dont la stratégie aléatoire, la stratégie de la limite supérieure de confiance de la bande multiarmée (MaB UCB1), la stratégie de la bande multiarmée ϵ -Greedy (0,1) et la stratégie bayésienne de la bande multiarmée (UCB1). Les résultats ont montré que l'algorithme Bayesian Multi-Armed Bandits UCB1 a fourni la meilleure récompense cumulative pendant la formation et a convergé vers les meilleures actions après 3 ans. L'algorithme ϵ -Greedy vient ensuite, suivi de l'algorithme MAB UCB1 et de l'algorithme aléatoire.

Le modèle de tutorat Selfit est conçu pour soutenir le processus d'apprentissage en s'adaptant aux caractéristiques du stagiaire grâce à l'apprentissage automatique. Le mésocycle de formation d'un stagiaire débutant est divisé entre les hommes et les femmes, en tenant compte des différences de cycle menstruel. Une simulation a été réalisée, montrant que l'application d'un plan d'entraînement standard pour les hommes aux femmes peut conduire à un entraînement sous-optimal, à des blessures ou à l'échec d'une séance. Les résultats suggèrent que des séquences d'entraînement spécifiques à chaque sexe devraient être envisagées.

Ensuite, l'expérience a comporté deux phases et deux versions du système. La version 1 de Selfit a été testée par 18 stagiaires en 2021 et les commentaires ont été utilisés pour l'améliorer. C'est ainsi que la version 2 de Selfit a été lancée en janvier 2022. La version actualisée a divisé les 42 stagiaires en deux groupes utilisant des algorithmes différents. Le premier groupe (groupe A) a utilisé un espace d'exploration plus large et le second groupe (groupe B) a utilisé un espace d'exploration plus étroit, avec l'espoir de meilleurs résultats pour ce dernier. L'expérience visait à valider l'architecture du logiciel et l'interface utilisateur et à évaluer si le système était efficace pour enseigner aux étudiants.

Les résultats ont montré que le RIR moyen du groupe A était plus élevé que celui du groupe B, ce qui signifie que le groupe A disposait d'un espace d'exploration plus grand et exposait des

récompenses plus petites en moyenne. Les valeurs du RIR pour le groupe B étaient plus proches de 0, et le groupe A avait des valeurs plus éloignées de 0 pour le premier mésocycle. Les résultats ont démontré la validité des tests d'étalonnage et de l'étiquetage de la difficulté des exercices. Les résultats du groupe A sont prometteurs et montrent une augmentation globale des récompenses. L'occurrence des récompenses élevées a également été évaluée et une augmentation progressive du nombre d'utilisateurs obtenant des récompenses élevées a été observée au fur et à mesure de la formation.

Enfin, l'étude a mené des enquêtes auprès des utilisateurs afin d'évaluer l'expérience de l'utilisateur de l'STI Selfit dans ses deux versions (Selfit Version 1 et Selfit Version 2). Les enquêtes ont utilisé le questionnaire AttrakDiff sur une échelle de Likert en 7 points pour évaluer les qualités pragmatiques et hédonistes du système. Les résultats de l'enquête sur la version 1 de Selfit ont montré que le système était généralement perçu comme pratique, prévisible, simple, connectif et orienté vers l'humain. Les résultats de l'enquête sur la version 2 de Selfit ont montré que les utilisateurs trouvaient le système plus agréable et plus personnalisé, avec des stratégies de formation améliorées. Dans l'ensemble, l'étude a montré que Selfit s'est amélioré grâce au retour d'information des utilisateurs et a fourni une expérience d'apprentissage agréable et efficace aux stagiaires.

10.2 Introduction

This chapter presents the experiments conducted to validate *Selfit*, both through simulation and within a real setup. Results include computing the Repetition In Reserve (RIR) per population – results at exercise-level (acute effect), and bandit algorithm learning over time – results at training program-level (chronic effect).

Also, the learning gains based on the pre- and post-tests are summarily presented, even though this assessment is not part of the overall goal of the experiment. Last, it presents the user experience results, for each phase, and a comparison between them. Both testing phases included a user experience survey, based on the AttrakDiff questionnaire (Hassenzahl, 2003), and it has shown that *Selfit Version 2* overall user perception was improved.

10.3 Simulation-based Experiment

The efficiency of the Tutoring module was simulated with different contextual multi-armed bandits' implementations. The experiments were conducted in fully virtual environments, and the goal was to determine, based on the general sports training methodology, which algorithm converges first, and how many training sessions are required.

The experiment was conducted in an environment configured with 1800 exercises, 300 for each movement family, with 10 exercises per level of difficulty. Difficulty levels ranged from 1 to 5. This first experiment did not take into account the sexual dimorphism dimension.

The current setup is closest to real use-case scenarios when compared to the current database of exercises. At a certain point, the tutor receives as input a movement family to train and must choose the right exercises, based on the internal load – an estimation of trainee level for the specific movement family and trainee state at that moment. Internal load is composed of the following parameters: resting heart rate, willingness to train, fatigue restoration state, sleep quality, and stress level. The following setup was considered for a person: 128 exercises per month, which means 32 exercises per week, with an average of 4 sessions per micro cycle, each of them with 8 exercises.

Four agents, each of them with different strategies were trained. The following strategies were implemented to assess performance:

- a) random agent – picks a random exercise for that movement type and level;
- b) multi-armed bandit upper confidence bound (MaB UCB1) – the principle of optimism in face of uncertainty, which means the more you are uncertain of an arm, the more important it is to explore;
- c) multi-armed bandit ϵ -Greedy (0.1) – explore (choose a random action) with probability ϵ and exploit (choose an action with maximum value) with probability $1-\epsilon$;
- d) Bayesian multi-armed bandit UCB1 – use the same principles of UCB1 but incorporate prior information on the distribution of an arm’s rewards to explore more efficiently.

For the ϵ -Greedy approach, the value of 0.1 for ϵ exhibited the best results for all the simulations. Initial competence levels were configured randomly for each movement type of the simulated trainees. The response of students after applying an exercise follows the standard Item Response Theory (Hambleton, Swaminathan, & Rogers, 1991), where the probability of being able to perform an exercise is given by Equation 3.

Equation 3. Item Response Theory (Hambleton et al., 1991)

$$p(\text{success}) = \frac{\gamma(a)}{1 + e^{-(\beta(c^Q - c(a) + \alpha))}}$$

Parameters β and α represent constants for simulating different learning rates of the population; $\gamma(a)$ was randomly generated for each competence level of the trainee between 0 and 1, where 0 means the trainee cannot perform the exercise. Also, it was considered that after every 30 exercises applied to a muscle family, the number of repetitions in reserve for all the exercises for the targeted muscle family will increase.

The experiment goal was to understand how fast and efficiently the proposed algorithms estimate and provide training exercises, using the current state of the trainee. In the experiment, a population of 1000 trainees was generated, each with a specific competence level, generated randomly per exercise; RIR values were initially estimated per exercise and trainee.

The previous population was trained for 2 years, including 384 sessions, using the four conditions given by the selected algorithms. The same exercises were applied to a trainee at a certain time, given the four strategies. Results can be seen in Figure 48. A data point on the Ox axis represents the current training session number, and the Oy axis encapsulates the cumulated training reward.



Figure 48. Training Algorithms Comparison – 2 years timeframe

The algorithm that provides the best cumulative reward during training is the Bayesian Multi-Armed Bandits UCB1: 1175; next was the ϵ -Greedy strategy (975.8), followed by simple MaB UCB1 (483.3) and random (33.1). Results after the first 6 months are close between the first 3 algorithms (session #48 in Figure 48); afterward, the Bayesian model constantly outperforms the others. Moreover, all the MAB are converging on the best actions starting in the third year of training. We can conclude that, for the first three years, the algorithm performing the best was Bayesian MaB UCB1, followed by ϵ -Greedy, and MAB UCB1.

10.3.1 Considering Sexual Dimorphism Dimension in Personalizing Training Sessions

The *Selffit* Tutoring model aids in the learning process by using machine learning techniques to adapt the learning program to the trainee's characteristics.

For the Novice coach, the sessions were generated using a standard calendar plan, used by trainers in their daily work, without integrating the gender dimension. Current work presents the calendar split between men and women, and the plans are computed in mesocycles (4-weeks long). The 28-day beginner load pattern differences can be seen in Table 23 and the example includes a man starting

the first day of training and a woman starting training on the same day as her first early follicular (menses) day.

Table 23. Mesocycle beginner load pattern differences between men and women

Man Load	L	L	L	L	L	L	L	M	M	M	M	M	M	M
Woman Load	L	L	L	L	L	M	M	M	H	H	H	H	H	H
Day in month	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Man Load	H	H	H	H	H	H	H	L	L	L	L	L	L	L
Woman Load	H	H	H	H	H	H	M	M	M	M	L	L	L	L
Day in month	15	16	17	18	19	20	21	22	23	24	25	26	27	28

For the days of the month which are shown as Light for man and Medium for woman (e.g., days 6, 7 in a month), if a woman would have been trained like a man, the session is considered sub-optimal. A sequence of sub-optimal sessions would produce sub-optimal training. This will imply more time to get the expected results and also the risk of losing trainee motivation, sessions may be too easy. On the other side, e.g., for day 21 a man would get a Heavy session, while an optimal one for the woman would be Medium – based on the cycle day, it is her first Mid luteal day. If the woman is trained based on a standard man plan, there is an increased risk of injury which may lead to session failure or quitting the training.

The training mesocycle presented in Table 23 is suitable for a beginner trainee; there are also 2 other types of mesocycle: development, more intense than the first, and, realization, which usually includes a challenge or a competition. In the realization mesocycle, most of the session loads are light, while the competition days have the biggest load. The mesocycle sequences are the same for men. The sequences are as follows for a beginner trainee: Beginning-Development-Realization and then they follow a Development-Realization loop: B-D-R-(D-R).

The importance of the sexual dimorphism dimension of tutoring is demonstrated through a simulation with hundreds of women having randomized menstrual days. A comparison between standard training and optimal woman training has been conducted and the accumulated loss was computed. Women and men are virtually trained for 240 days, based on 3 sessions per week plan.

The optimal sessions load for a beginner woman and a beginner man starting on the same day can be seen in Figure 49 a) and b). The 1-5 values on the axis show the session load type: 1 – Light, 2 – Medium, 3 – Heavy, 4 – Very Heavy, 5 – Challenge session.

The bars in Figure 49 c) show the difference between men’s and women’s training; for example, the first blue bar shows a man’s session load equal to 2, and a woman’s session load equal to 1. The sessions with blue bars show these have a risk of injury and potential session failure. For the presented simulation, the accumulated risk of injury loss was -49.

The sessions with no associated color (value zero) show that the proposed session is optimal – matching men’s and women’s optimal load. The overall count of loss in sessions (values different than zero) is 82, whereas the cumulated differences are -16. This shows that man training planification applied to women is sub-optimal and can lead to injuries; thus, gender-specific training sequences should be considered.

Understanding the mechanics of motor skill acquisition is critical for designing an ITS for training psychomotor capabilities. Such a smart learning environment should also consider the sexual dimorphism dimension when training for motor skill acquisition, the differences being demonstrated by several morphological and physiological studies.

Several experiments were conducted in a fully simulated environment, where populations of trainees were generated, with corresponding exercises of different characteristics. Four teaching strategies were tested, from which Bayesian MAB UCB1 exhibited the best results for a scenario close to a real-world use case.



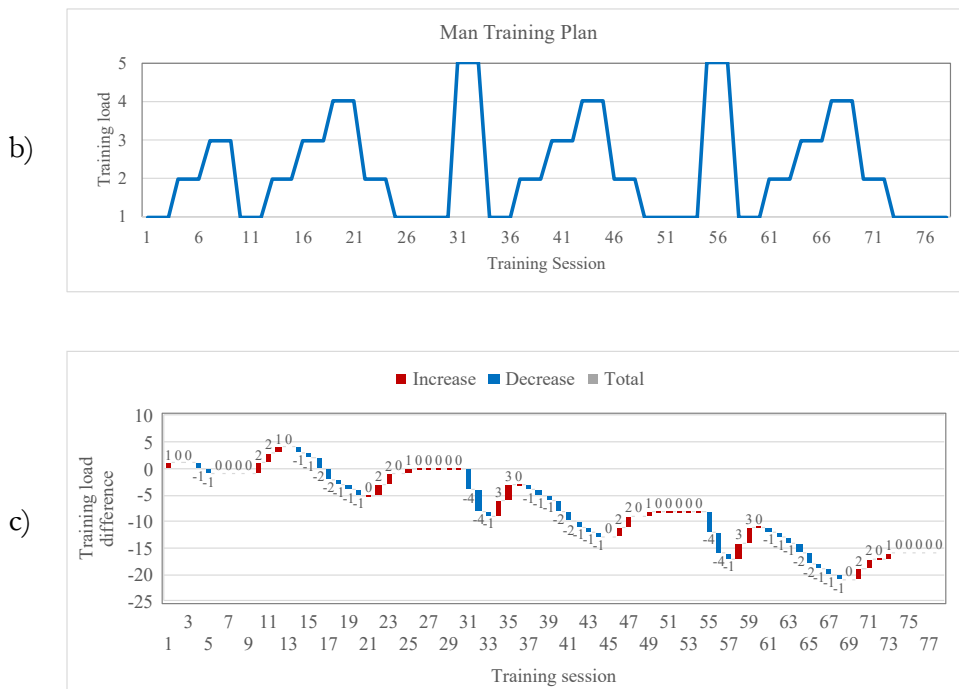


Figure 49. Differences in Training Planification considering Sexual Dimorphism.

An improved version of the initial system is presented, where the tutoring module uses men’s and women’s mesocycle calendars, based on the menstrual date and periodicity. Standard training plans use the same training rules for both men and women, while sports science has proven the contrary, with risks of suboptimal training, injury, or training failure. As the simulation shows promising insights, future work should follow up the contribution to the smart learning environments used for psychomotor skills development. In the future, the monitoring of the morphological and physiological trainee body parameters, while performing personalized training, may provide valuable insights, which can open new research directions for sports science.

10.4 Experiment with Real Users

The goal of an Intelligent Tutoring System is to provide more efficient teaching experiences to students (Clement et al., 2015). The current experiment aimed to evaluate both *Selfit* learning improvement and the overall user experience. The experiment has been split into two phases. *Selfit version 1* was initially tested between January-February 2021, by 18 trainees from France and Romania. Few user interface and session exercise selection bugs were reported by the users, and several improvement features were proposed in this preliminary feedback.

The initial testing phase has been followed by a new development phase, between February 2021 and December 2021, for an improved *Selfit* version, which was rolled out to production at the end of 2021. *Selfit Version 2* included several bug fixes, performance improvements, and a set of new

features, based on user feedback. Features include the ability to customize your profile (set up a profile picture, set a motto, birthdate, etc.), an option to pause and resume time-based exercises, an improved interface for inputting RIR per exercise, better tailoring of a session by inputting a list of materials available to train, added body areas trained statistics diagram, integrated Google Analytics, created a new protocol for pre- and post-tests, and others.

Selfit Version 2 was tested in the second phase of the experiment, between January-May 2022, by 42 trainees from France and Romania, which got onboarded in the app on different dates. Even though the system has been promoted among several social channels and 77 accounts have been created and have logged in at least once in the system between January and May 2022, only 42 users have performed the pre-test and trained at least once using the *Selfit version 2* system.

The system was further evaluated with real users, from both Romania and France, and the goal was to validate the software architecture, and the interface, and to assess if it is possible to learn the best load over each exercise, so that learners are at their optimal level across a training program. The tutoring module used the Novice trainer approach, where the adaptation is made at the exercise level. The sessions and microcycles were generated based on pre-defined templates with generic exercises and areas to train – as it was described in 8.4.

The users involved were recruited from the close group of *Selfit* researchers, and were mainly either software developers, students, teaching assistants, or sports athletes. The system has been published in the mobile stores (Android and iOS) and made available across all countries. No organic reach from the mobile stores was shown in the statistics. The system has been disseminated across our research group channels – initially, we sent out a mail within our research group – "Centre de recherche sur les risques et les crises (CRC) MINES ParisTech", Sophia Antipolis, to up to 100 people, where we described our work and then we disseminated the *Selfit* system to our social channels – we made several posts asking people to enroll in the program which were shared across our LinkedIn¹⁸, Facebook¹⁹, and Instagram²⁰ profiles.

Most of the recruited participants were between 20-40 years, and the majority was made up of males. The experiment was split into two phases, and it covered two versions of the *Selfit* system, as follows:

- *Selfit Version 1*, tested between January-February 2021, with 18 trainees, which outlined a list of usability bugs and recommendation flaws;

¹⁸ <https://www.linkedin.com>

¹⁹ <https://www.linkedin.com>

²⁰ <https://www.instagram.com>

- *Selfit Version 2*, tested between January-May 2022, with 42 trainees, including the group involved in *Selfit Version 1*, which has shown the improvements from version 1 and the value of Multi-Armed Bandits algorithms in psycho-motor tutoring.

Even though the first phase of the experiment should have lasted for 3 months, to also gather data about how the tutoring module behaves in practice, based on the user's feedback and the bugs reported, this phase has been paused and the issues were addressed. *Selfit Version 2*, with several interface and performance upgrades, has been released internally, in our research group, in November 2021, and released publicly, in the mobile stores, in January 2022.

The updated version has split the population of trainees in two, both using Contextual Multi-Armed Bandits ϵ -Greedy (0.1) algorithm:

- the first group has used the *Selfit Tutor A* – the tutor is using a wider exploration space (the bandit arms for a movement area exercise include all the available levels, filtered by the available materials);
- the second group has used *Selfit Tutor B* – the tutor is using a narrow exploration space (the bandit arms for a movement area exercise include only the user's estimated level, filtered by the available materials).

We expected the *Selfit Tutor B* to provide more tailored training content overall, due to the smaller size of the exploration space. Even considering the current trainee context, which we assumed it should impact the training load – difficulty of the exercises provided, we expected more engagement and better progress for the participants in Group B.

Based on Google Analytics statistics, more than 50 people have tested the application for several weeks during the experiment, e.g., for the week 29th April – 5th May, as can be seen in Figure 50. For this specific week, it is also shown that there is an average of 15 trainees in the app each day, with the lowest number on 30th April (10 users) and the peak on 3rd May (19 users). The previous week, which is marked as a dashed line in the figure, can be seen it had similar values per day and the average value was also close to 15 trainees per day. Even though there were several weeks with 50 or more users tracked by the Google Analytics, only 42 trainees were included in the experiment. The minimum criteria to fit was to create a profile in the system, perform the initial calibration challenge, and complete at least one session until end of May 2022.

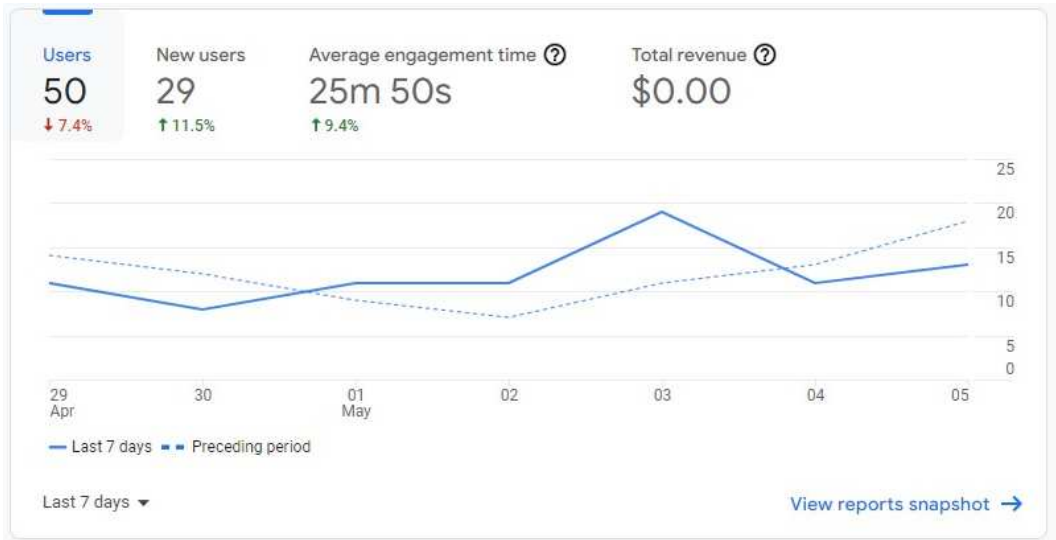


Figure 50. Google Analytics Statistics. Users in *Selfit* between 29th April and 5th May.

Figure 50 also shows the average engagement time for the first week of May 2022, which is 25 minutes and 50 seconds. Even though the current training sessions currently have fixed lengths: 30, 45, 60, 75, and 90 minutes, Google Analytics measures any user authentication in the system as an engagement. As the users did not always log in to the system to train, they also just checked their past sessions, updated their profile, checked their statistics or other users, and the reported average time spent in *Selfit* has overall an expected value.

The trainees have used *Selfit* every daily during a week, and in most of the available timeframes of the day, as can be seen in Figure 51. The weekday with the most sessions performed was Tuesday, where three slots had close to 80 training sessions performed: 5 pm, 6 pm, and 9 pm. The time shown in the figure was computed based on France’s local time, namely GMT+2. It can be seen also that at 3 am no sessions have been performed on any weekday or weekend, while for midnight only on Friday one session has been completed.



Figure 51. Training Sessions Count per day and hour slot.

Users enrolled in the experiment were exposed to one of the two versions of the tutoring component, both implemented using the Contextual Multi-Armed Bandits ϵ -Greedy 0.1 (Contextual

MaB) session generation – one with a higher exploration space (called further Group A), and another one, with a lower exploration space (called further Group B).

They were assigned to each group based on the order they registered within the system. After registration, the system generates an incremental unique identifier. The odd numbers allocated the users in Group A, and even numbers allocated the Group B users. The 42 users involved in the experiment were split as follows: 22 in Group A, and 20 in Group B. Even though the split per group was even, the number of exercises performed was unbalanced between the groups. The total number of exercises recommended by the bandit in this phase of the experiment was 1521, from which Group A performed 1297 (85.27%) and Group B performed 224 (14.73%). This was a surprise for us considering the experiment setup described previously. We expected a similar engagement between the two groups, and even Group B to surpass Group A in terms of overall training time. We will try to better understand the discrepancy between the two groups as we will look in detail at the training data.

A plot of the exercise count executed in the most active training slot of phase two of the experiment, 11th April to 23rd May, can be seen in Figure 52. The first part of a training session is the warmup, with a length of approximately 10 minutes. The warmup has different exercises based on the session body target and includes on average 15 exercises, to increase the athlete’s body temperature and avoid the risk of injuries while training. Warmup exercises do not require any feedback from the user, so no RIR is computed, all the exercises are time-based and were not tracked in the experiment.

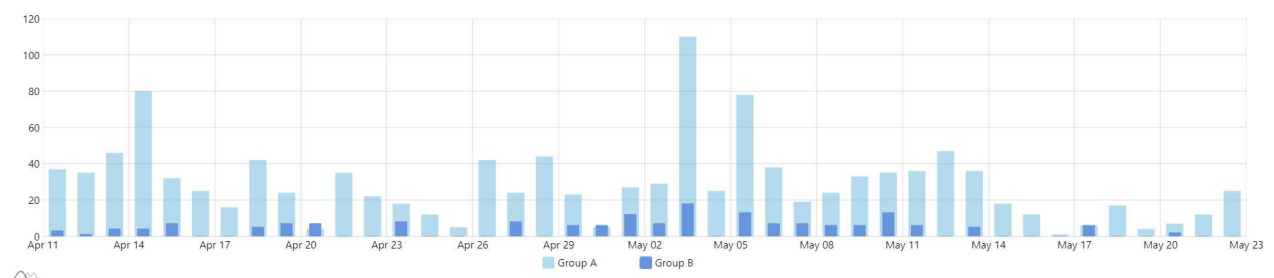


Figure 52. Exercises count per training group between 11th April – 23rd May.

Based on the user’s feedback while training – some users were using their custom warmups to increase body temperature, or others have complained *Selfit Version 2* warmup was too long and there were too many buttons to press to get to the content part of the session. An option to "Skip the warmup" have been developed and published at the end of April within the system.

The second part of a training session is the content or the core, which was generated based on the target of the current session within the microcycle, the location of the current training (at home or gym), the available time (from 30 to 90 minutes), the available materials, and a list of muscles

preferred to train. The number of exercises in the second part of the session is determined by the location and the time availability. The mapping is presented in Table 24. Only content exercises were analyzed in the current experiment, which covered in total 315 training sessions.

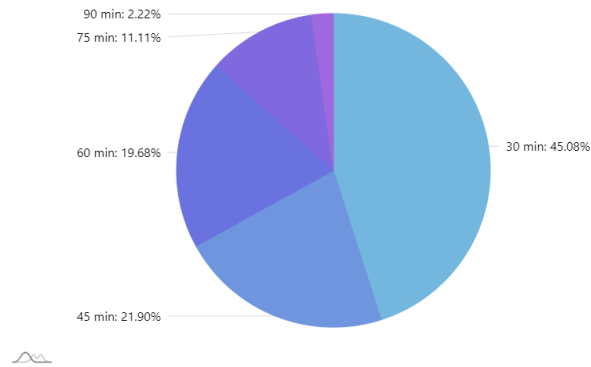
Table 24. *Selfit Version 2* mapping of Time Availability and Count of Exercises Proposed.

Location	Time Availability (minutes)	# of Content Exercises
Home	30	5
	45	5
	60	6
Gym	60	6
	75	7
	90	8

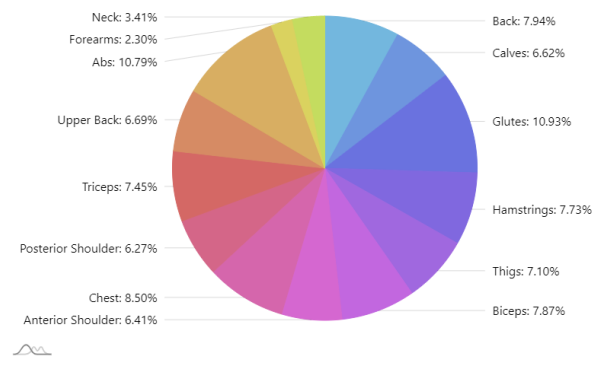
The top 5 performers from Group A have the following number of sessions completed: 30, 27, 25, 23, and 16, while from Group B the top 5 performers have: 11, 6, 6, 5, and 5 sessions respectively. Group A has performed 254 training sessions, while Group B has performed 61 training sessions.

Even though the split among groups has been performed based on the random-like order of users joining the system, the statistics show the top 3 performers from Group A have trained more than all the trainees in Group B (20 users).

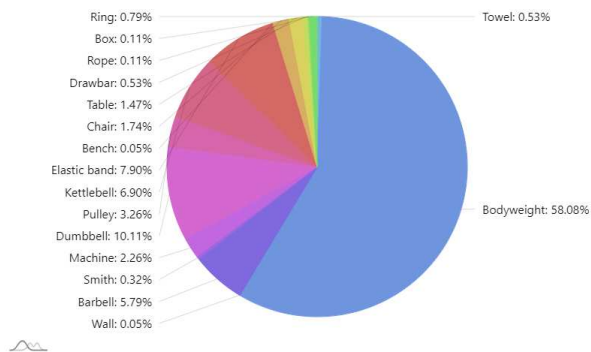
Most of the time users have chosen 30-minute sessions at home – 142 times, followed by 69 training sessions with a 45-minute length at home. Then, there are 62 sessions completed in a 60-minutes length, both at home and at the gym. 35 sessions, 11.11% of the total completed training session, have been completed at the gym and within a 75-minute allocated time. The remaining ones, 7 sessions, out of the 315, were performed with a 90-minute time slot, in a gym setup. The statistics of train time chosen per session can also be visualized in Figure 53 (a). The split between home and gym sessions was the following – 225 training sessions performed at home (71.43%), and 90 training sessions at the gym (28.57%).



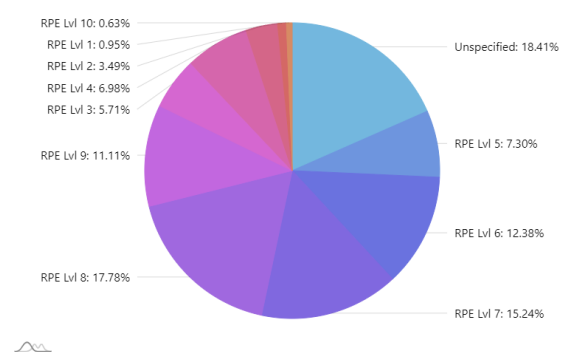
(a) Train Time per session



(b) Muscles to train per session



(c) Materials required in exercises provided



(d) RPE Level reported by users per session

Figure 53. Experiment Statistics on Train Time (a), Muscles to train (b), Material used (c), and RPE (d)

Even though each session has a specific target, with a list of muscles to focus on, based on the microcycle or weekly planification before the start of the session users are requested to choose a list of complementary muscles to train. The list of available muscles to select was identical between home and gym training and users could have picked as many muscles as they want, or none. The experiment showed the choice of users was diversified, many muscles were chosen by users many times, see also Figure 53 (b). All the muscles available in the list, 14 in total, were chosen at least once by the users enrolled in the experiment. The top choices were glutes (157 times), abs (155 times), chest (122 times), back (114 times), and biceps (113 times). On the lower end, choices were forearms (33 times), and neck (49 times), while the remaining ones were close to 100 selections.

A list of available materials to train can be configured also at the beginning of each session, and this affects session generation by providing a wider range of exercises to pick from. When no material is selected by the user, by default only bodyweight exercises are selected. The material lists are different between home and gym training, only elastic band, kettlebell, and bodyweight are shared between the two locations. Top users' choices of materials for home training were bodyweight (137 times),

chair (72 times), elastic band (54 times), kettlebell (41 times), and table (39 times). When configuring a session in the gym environment, users selected to use the dumbbell (84 times), barbell (80 times), pulley (63 times), body weight (58 times), or machine (57 times).

The choice of materials impacts the exercises selected for each session. There are exercises where either one or other material can be used to execute, or two materials are required. The exercises chosen, which are based on the session configuration and user profiles, were in a vast majority bodyweight exercises – 1103 exercises, followed by exercises that required a dumbbell – 192 times, then an elastic band – 150 times, and kettlebell – 131 times.

On the lower end, the wall has been required 1 time, Smith 6 times, or ring 15 times. The statistics showing the list of materials required by the users for practicing the exercises recommended in the experiment can be seen in Figure 53 (c).

The *Selfit* system records the immediate feedback for each exercise, represented by the RIR per set, and per side (left and right) for unilateral exercises, and the exercise-level feedback will be described further in II.2.2. System also asks for the user's input on the overall perception of a session difficulty, called Rate of Perceived Effort (or RPE), on a 10-Level scale, where 1 is perceived as very hard session, and 10 as very easy. Users were not required to input the RPE field. Out of the 315 sessions, for 58 sessions there was no RPE specified. The most voted value among all sessions was Level 8 (56 times), followed by Level 7 (48 times) and Level 6 (39 times), which means the sessions were mostly perceived as average – we considered values greater than 8 as easy sessions, and values less than 5 as challenging, so we have 166 sessions (64.5% of RPE reported) marked between 5 and 8. On the other end, 3 sessions were voted with Level 1 (very hard), and 2 voted with Level 10 (very easy). The statistics with levels for RPE and the number of sessions associated can be seen in Figure 53 (d).

1479 exercises out of 1521 initially recommended have been displayed to the users in *Selfit Version 2*. For the rest, the user stopped the application with an ongoing session and did not finish it until 31st May, so no data has been recorded. Each exercise described in the Domain model involves the movement coordination of one to four joints which, in some cases, is also linked with the exercise difficulty, as described in II.7.3.2, OntoStrength structure. The exercises proposed in the experiment had the following split, based on the number of joints: one joint (421 exercises), two joints (299 exercises), three joints (680 exercises), and four joints (79 times). Exercises are also categorized as unilateral and bilateral. This implies that the number of repetitions and number of sets per exercise should be executed separately on the left and the right (unilateral exercise), or at the same time

(bilateral). The current experiment recommended 1276 bilateral exercises and 203 unilateral exercises.

The team of sports scientists who have worked on *Selfit* ITS development also labeled the difficulty of an exercise and recorded or got access to a good quality recorded video for each exercise which was shown to the user, before the execution.

Exercises were labeled initially on a 12-level scale and then grouped into four categories – Beginner, Intermediate, Advanced, and Expert. These categories are also matching the calibration challenges levels, useful to group users per each movement type in the corresponding category. The statistics with the number of exercises recommended per each difficulty level can be seen in Table 25. Most exercises recommended were at Level 3 (341), followed by Level 6 (240) and Level 5 (215). The least number of exercises were provided from Level 10 (9), and Level 7 (44). By categories, the ranking was the following – Beginner (636 exercises), Intermediate (512 exercises), Advanced (216 exercises), and Expert (115 exercises).

Table 25. Count of exercises recommended per exercise level.

Exercise Category	Exercise Level	# of Exercises	Exercise Category	Exercise Level	# of Exercises
Beginner	1	85	Advanced	7	44
	2	210		8	112
	3	341		9	60
Intermediate	4	57	Expert	10	9
	5	215		11	54
	6	240		12	52

Top exercises selected and provided to users were – sliding pull up on the floor (49 times), push back (36 times), wall pull (18 times), push up (15 times), modified inverted row (14 times), shoulder elevated hip thrust (14 times), and Romanian deadlift with knee raise (14 times). On the other end, some exercises were not picked at all, due to materials mismatch, or user level constraints, and there were many exercises that were selected only once or twice.

The calibration challenges' goal was to classify the user in one of the four-level categories (Beginner, Intermediate, Advanced, or Expert), per each movement type: Upper Body (Push Horizontal, Push Vertical, Pull Horizontal, Pull Vertical) and Lower Body (Hip Dominant, Knee Dominant). Users can update their levels per movement type only by performing again the challenges, at any moment. Based on the calibration test, users are assigned in the corresponding level per movement type and future exercises will be provided following the current values.

Bandit's recommendations are different between the two groups. Group A, which uses the high exploration space, will have more available arms to choose from – the bandit will choose from all the levels of that movement area, and be filtered by user restrictions and preferences. Group B, which uses the narrow exploration space, will have fewer available arms to choose from than Group A. Here, the bandit can choose from the exercises mapped at the corresponding user level for that movement area, also filtered by restrictions and preferences, which include available materials and muscles preferred to train. More details of how the session templates are filled in and on the mapping rules of movement areas with body areas and muscles can be seen in 9.3.1 and 9.3.3.

Tutor B's strategy was to use a smaller exploration space, but it may prove suboptimal considering the variability of user states - e.g., when the user is tired – simpler exercises may prove to be better, or when a user is in great shape it's possible that harder exercises to be more efficient. Group A is targeting all levels, from Beginner to Expert, for each movement type and the bandit algorithm learns from experience, considering different user states. An example of how the algorithm behaves for two users, with the same training profiles, who are using Tutor A and Tutor B, after the calibration challenge, can be seen in a simulated scenario in Figure 54.

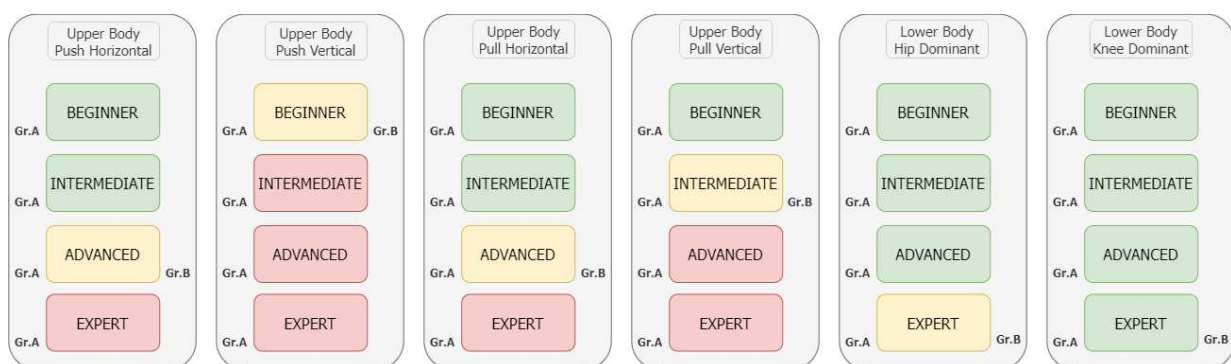


Figure 54. Bandits Exploration Space per each group, based on the Calibration Challenge.

For the Upper-Body-Push-Horizontal movement type, the current level is Advanced. In Group A (Gr. A), exercises provided can be either Beginner, Intermediate, Advanced, or Expert, while in Group B (Gr. B), the exercises will be only from the Advanced category.

10.5 Results of Agent's Learning

The experiment's goal of learning was twofold. First, we aimed to validate the calibration challenges protocol and the sport exercises classification. For this, Group A and Group B followed the same calibration tests and were classified within corresponding groups. Considering the higher exploration space for Group A, we expected, at the beginning of the training program, to expose smaller rewards on average than for Group B. RIR values were expected to be closer to 0 for Group B, and farther from 0 for Group A for the first mesocycle.

Second, we aimed to validate the efficacy of the Contextual Multi-Armed Bandits algorithm on learning. For this, the average rewards at each step were computed in each group and the bandits' learning is discussed. This phase also presents the potential noise in a few athletes' data and describes an exclusion protocol for faulty data.

10.5.1 Results for the Acute Effect – Exercise Level Feedback

The 42 trainees involved in the experiment, split among groups – 22 in Group A, and 20 in Group B, practiced in 315 training sessions. A more detailed view of the number of sessions per user and its group can be seen in Figure 55. We can conclude that Group B had many light trainees – 9 users with only one training session and 2 with two training sessions, while Group A had many heavy trainees – 12 users with more than 12 sessions, and none in Group B.

The median on the array with all the sessions per user has been computed (the value was 5), and a significant sub-array has been selected, with users who performed sessions between 2 and 11. At this stage, the measurement of RIR as an acute effect of an exercise aims to compute the difference of the two groups, to validate the calibration protocol accuracy and the labeling of exercises' difficulty.

Assuming the testing protocol was correct, users are assigned to the right levels per movement type, and exercises were labeled correctly, we would expect Group A, having a larger exploration space, to provide more inputs on average with very hard or very easy exercises than for Group B, for the first part of a training program.

We consider as relevant for demonstrating the validity of the calibration tests and labeling of exercise difficulty for the users who trained between 2 and 11 sessions (1 session \approx 6 predictions), as for the ones with more than 12 sessions, the bandit algorithm starts to gain experience and the rewards are expected to start converging between the two groups. We also excluded from the significant sub-array the users with only one training session, as the data for them has the risk of being noisy. The filtered data, the array [2, 11] – also called the Significant array, includes 8 users from Group A, who

performed 121 exercises, and 11 users from Group B, who performed 165 exercises. As can be seen above, this array has well-balanced data between the two groups.

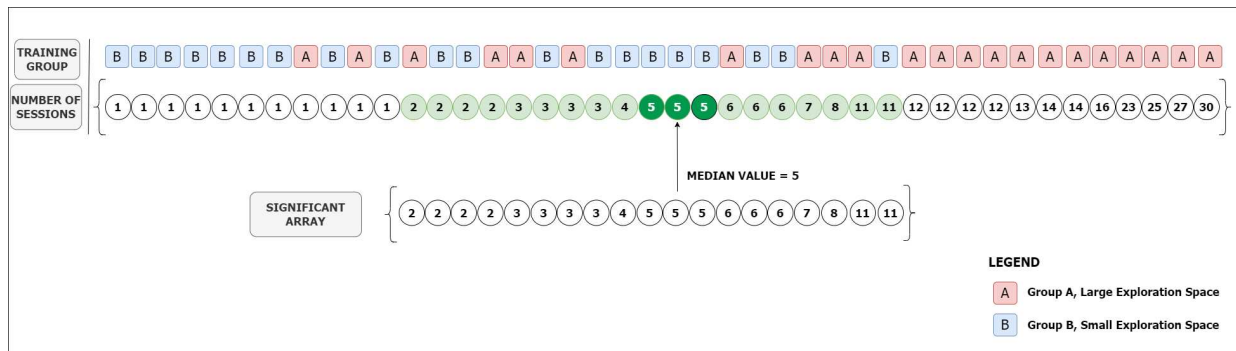


Figure 55. Number of sessions per user and their corresponding training group.

The experiment included 1479 exercises that users have seen and interacted with while training. From those, users have filled in at least one value for RIR for 1442 of them. Initially, *Selfit Version 2* allowed users to move throughout the training session without filling in any RIR value. A hotfix has been released at the beginning of May which made the RIR field mandatory. 37 entries have no value for RIR and were excluded from the current analysis. For the significant sub-array, 21 entries had no RIR specified, and these entries were not used further.

An exercise provided for a user at a certain time has between 3 to 8 RIR input values to fill in. 3 inputs are the minimum – specific to 3 sets, bilateral exercise – see Figure 56 (a), while 8 is the maximum - 4 sets, unilateral exercise – required to input RIR for each part, left and right – see Figure 56 (b).

The allowed values for RIR are:

- -10, which means "I performed less or equal than (number of reps required - 5) "; e.g.: for 12 reps, it could not perform more than 7;
- -4, which means "I performed less or equal than (number of reps required -4) ", e.g.: for 12 reps, I only performed 8 repetitions);
- -3, same logic as for -4, value is (number of reps required -3);
- -2, same logic as for -4, value is (number of reps required -2);
- -1, same logic as for -4, value is (number of reps required -1);
- 0, which means "I was able to perform the number of reps requested, but I am not able to perform any other";
- 1, which means "I was able to perform the number of reps requested, feel like I can perform one more". Value is (number of reps required + 1);

- 2, same logic as for 1, value is (number of reps required + 2);
- 3, same logic as for 1, value is (number of reps required + 3);
- 4, same logic as for 1, value is (number of reps required + 4);
- 10, which means "I was able to perform number of reps requested, feel like I can perform more than the number of reps + 5".

For simplifying the interpretation of the statistics, the absolute value of the RIR has been considered in this stage. An optimal value for the RIR is considered a value that is close to zero. The RIR was computed both per each set performed, but also at the exercise level (sum of all sets in an exercise).

The average RIR computed per set and per exercise for each experimental group can be seen in Table 26. The first line in the table covers the total amount of data from the experiment, interval [1,30], and shows that Group B outperforms Group A on both average RIR per set (34% smaller), and average RIR per exercise (26% smaller).

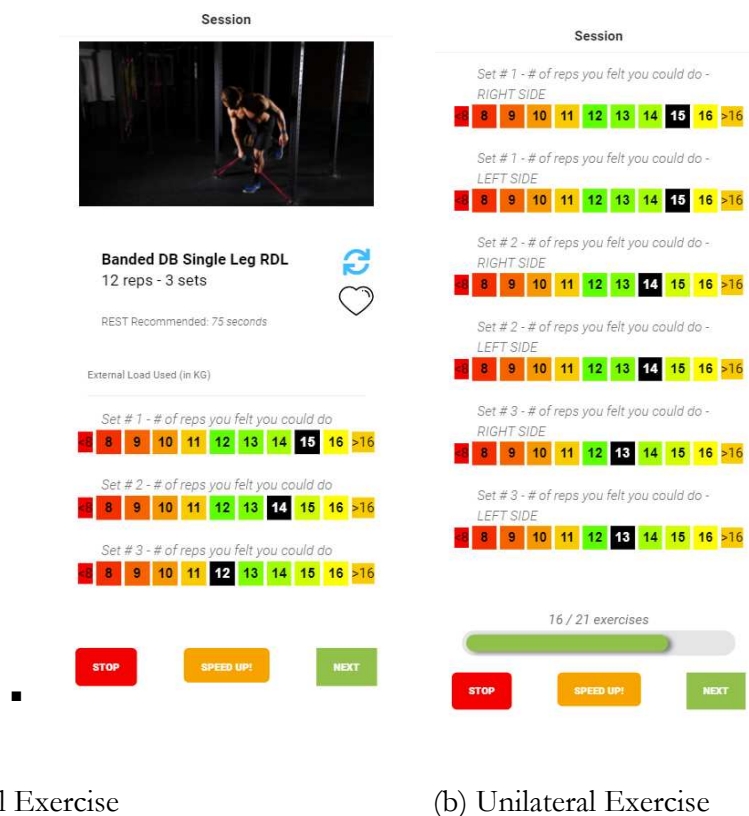


Figure 56. User Interface for Inputting RIR per Exercise type

The values are even more clear on the significant array, [2,11], where Group B has an average RIR per set of 2.63, as compared to 5.93 in Group A, and 8.08 is the average per exercise in Group B, while in Group A is 14.84. The significant array can be considered the most representative of the experiment, as the selected data is relevant and balanced between the two groups.

The overall results obtained by analyzing the average RIR can be justified by the fact that people who trained for 11 sessions or less in Group A have received easier or harder exercises, and in the long run, after 12 sessions, this value is starting to decrease. This is because bandit algorithms start to provide better exercise recommendations, as they are learning from experience, as will be discussed further in the next section. It can be also seen that while comparing lines one and two for Group A, where the values are decreasing as the number of sessions per user increases. For up to 11 sessions, the average RIR per exercise is 14.84, while for all the data (up to 30 sessions), the average RIR per exercise is 11.34. A similar difference can be also seen in the average RIR per set.

As it can be seen, all the other sub-arrays presented in Table 26 have shown a smaller average RIR over Group B than in Group A. For the users who performed fewer sessions, the difference is even more significant - see line 2 -array [2,11] -, and line 3 – array [3,8] - in the table. As the users trained more, the difference is decreasing. We can observe a difference between intervals [3,16] and [3,30] in Group A.

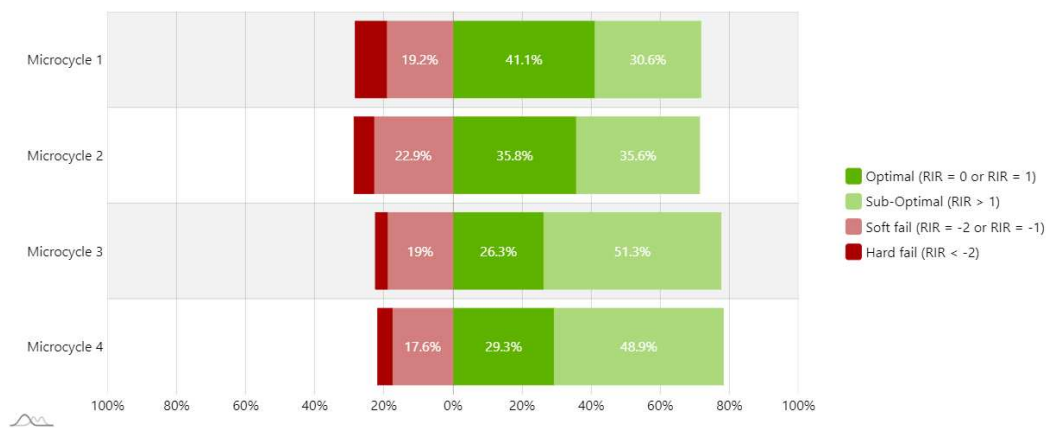
The ones who have trained less, in the first interval, have an average RIR per exercise of 12.89, as compared to the second interval, where the value is 11.27. The average RIR per set is also 18% smaller for the second interval. For the trainees who are using the system consistently, it was shown that the average RIR in Group A tends to decrease over time and converges with the one in Group B. The last line of the table shows that for the four users who used the system for more than 23 sessions, the average RIR per exercise is 9.18, while the average RIR per set is 2.94, close to the values in Group B, in fewer sessions.

Table 26. Average RIR per exercise and per set with different sessions interval sub-arrays.

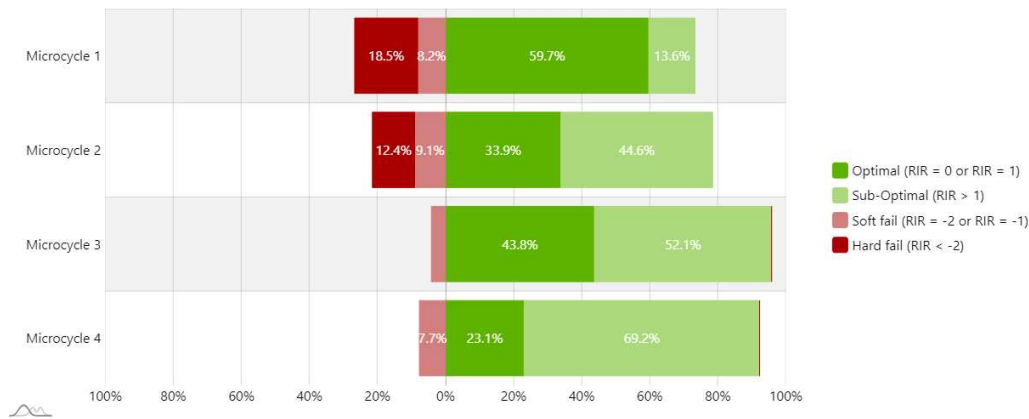
# of Sessions Interval Array	# of Users / Exercises Group A	# of Users / Exercises Group B	Average RIR per exercise Group A	Average RIR per exercise Group B	Average RIR per set Group A	Average RIR per set Group B
[1, 30]	22 / 1250	20 / 192	11.34	8.38	4.14	2.70

[2,11] (Significant Array)	8 / 121	11 / 165	14.84	8.08	5.93	2.63
[3, 8]	5 / 53	8 / 94	15.15	8.05	5.94	2.38
[5,12]	8 / 362	6 / 120	13.70	9.76	5.65	3.24
[3, 16]	14 / 688	9 / 151	12.89	8.65	5.04	2.80
[3, 30]	18 / 1218	9 / 151	11.27	8.65	4.13	2.80
[23, 30]	4 / 530	- / -	9.18	-	2.94	-

This analysis considers, at each subinterval ([2, 11], [3, 11], etc.), all the user's sessions, without considering the current step in the training program. We were able to see, generally, that for users who trained more, the average RIR per session and exercise tends to decrease over time. To see if there was any improvement in RIR while training, a more granular analysis has been performed, per training microcycle - all the first microcycles for the users in Group A and Group B, all the second microcycles, and so on. The results for the first four microcycles per training group can be seen in Figure 57.



(a) Group A – Optimal / Sub-Optimal / Soft fail / Hard fail RIR values for the first mesocycle



(b) Group B - Optimal / Sub-Optimal / Soft fail / Hard fail RIR values for the first mesocycle

Figure 57. RIR values in first mesocycle for (a) Group A and (b) Group B trainees

Figure 57 (a) presents the first four weeks of training for Group A. The differences between weeks of RIR percentages are not significant, we see though a trend for hard and soft fail RIR to decrease. Week 1 is 28.2%, while in week 4 is 21.7%. Figure 57 (b) shows the RIR percentages in the first month of training for Group B. The hard failure RIR is decreasing after each week, and becomes 0 starting in week 3, and this can be justified by the smaller space of exploration for Group B. Also, the overall cumulated optimal and sub-optimal RIR tends to grow as the users advance in the training program. As for comparing the differences in positive RIR between the two groups, we can see that in each microcycle the values in Group B surpass the values in Group A: microcycle 1 (A – 71.1%, B – 73.3%), microcycle 2 (A – 71.4%, B – 78.5%), microcycle 3 (A – 77.6%, B - 95,9%), microcycle 4 (A – 78.2%, B – 92.3%). We can conclude that Group B, with a narrow exploration space, provides better RIR on average than Group A in the first 4 weeks of training – first mesocycle. It is also important to note that the Hard Fail was also higher in Group B in the same timeframe. This will be discussed further in the next sections.

Group B had no trainee who followed the program for more than one mesocycle. In Group A, there were 8 users which performed more than four weeks of training (12 training sessions) and we observed the percentages for optimal and sub-optimal RIR in the second microcycles tend to grow. For example, in Microcycle 6 the cumulated value was 87.1%, while in Microcycle 8 the cumulated value was 85.6%. Even though the values are not continuously increasing (Microcycles 5 and 7 have similar values to the ones in the first mesocycle for the same users), it is a clear tendency of the bandit algorithms to start gaining knowledge and to provide better recommendations. This will be explored in more depth in the following section.

The acute effect at the exercise level is different between the two groups. The first mesocycle in the training program shows, on one hand, more optimal and sub-optimal RIR values on average in

Group B than in Group A, which means the calibration tests are classifying the trainees in the right categories. If the classification was not correct, more failures or less optimal values would have been shown in comparison with Group A. On the other hand, the labeling of exercises difficulty in the *Selfit Version 2* system is generally perceived as correct based on the results in Group B. Even though for a specific movement area level user can receive exercises that gets him to fail – see Microcycle 1 in Group B (26.7% of both soft and hard fail), this number of exercises is limited, and the majority in that specific level is labeled correctly. We observe high values for both optimal and sub-optimal RIR for Group B. Based on all the above considerations, we can conclude the results obtained in the experiment validate the training testing protocol and the labeling of exercises difficulty.

10.5.2 Results on Bandits' Learning

The second goal of the current experiment on learning was to validate the efficacy of the Contextual Multi-Armed Bandit algorithms in sports training planification. Top trainers from both groups have been analyzed in this phase, as they had the most predictions steps to assess the learning.

Group A, characterized by a wide exploration space from the bandit, required more states than Group B to provide a better reward across time. We considered it relevant in Group A, based on the simulations presented in 10.3, users who performed more than 12 sessions. The selected interval included 12 users and had from 57 to 169 actions taken by the bandit based on the current trainee state. The results of the average rewards for top performers in Group A can be seen in Figure 58. The figure presents each bandit prediction step, from 1 to 144, and the average reward across users tends to increase over time.

Even though the most active trainee in Group A trained for 169 steps, we considered in the current analysis the averages of at least three trainees at each data point. This is the reason the number of bandit steps in Figure 58 stops at 144, the last step when there were three users with predictions. The information with all the top trainees in Group A, their corresponding bandit steps, and the number of sessions can be seen in Table 27. We can state that, at each User ID, the number of trainees at that bandit step is equal to the value of that identifier (value of User ID). For example, in step 153, 2 trainees were using the system, and starting step 154, only one remained.

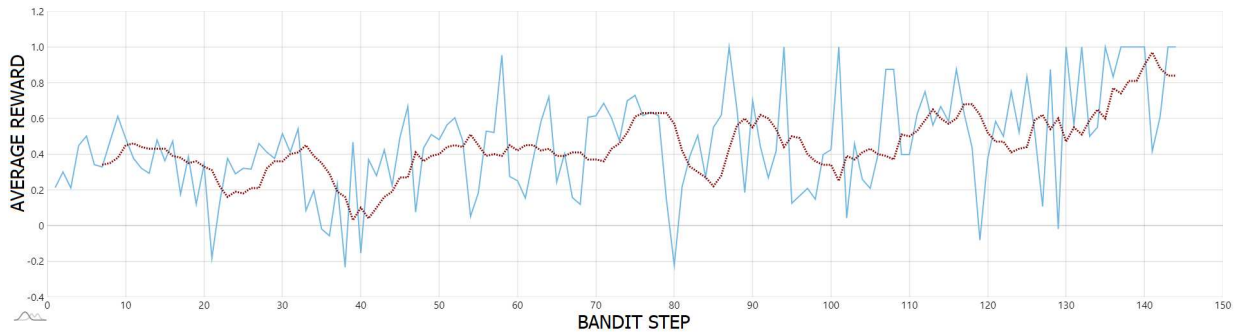


Figure 58. Average Rewards and Moving Average (window size = 6 steps \approx 1 session) Rewards per Bandit Step for Top Performers in Group A (> 11)

Same for bandit step 78, 8 trainees were using the system, and starting step 79 to step 81 there were 7 trainees, and so on. The number of sessions is also displayed in the table, and the values are not all the time in descending order, as it is for the bandit steps.

Table 27. Group A Top Trainers (≥ 12 sessions) Bandit Actions and Sessions Performed.

Top User ID	# of Bandit Steps	# of Sessions	Top User ID	# of Bandit Steps	# of Sessions
#1	169	27	#7	81	13
#2	153	30	#8	78	12
#3	144	23	#9	73	14
#4	135	25	#10	70	14
#5	93	16	#11	70	12
#6	84	12	#12	57	12

This is due to the length of specific sessions. For example, User ID #1 with 27 sessions had longer sessions than User ID #2, with 30 sessions. Longer sessions use training session templates that have a bigger number of exercises, so the number of agent predictions is bigger.

Each session generates, on average, 6 agent actions. Figure 58 shows also, on the dashed dark red line, the moving average which used a window of 6. The general increase of the reward over time is also easier to be observed with the moving average, at the average session length window.

Similarly, for group B we computed the average reward across all trainees at each prediction step but, due to the limited data, we looked between 3 to 11 sessions, which covered a number of 6 trainees, and considered in our analysis steps where at least 2 trainees are involved. The results for Group B can be seen in Figure 59. The bandit steps for the 6 trainees and their corresponding sessions are the following, in descending order: 70 steps (11 sessions), 38 steps (6 sessions), 32 steps (5 sessions), 31 steps (5 sessions), 20 steps (4 sessions), 18 steps (3 sessions), and 17 steps (3 sessions).

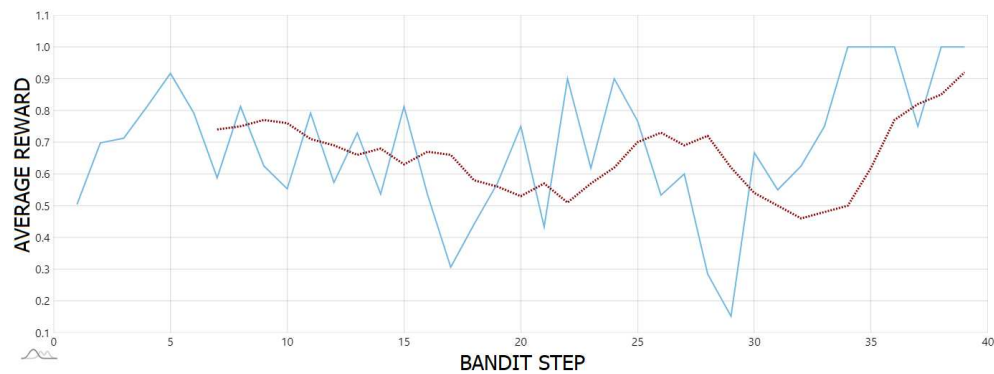


Figure 59. Average Rewards and Moving Average (window size = 6 steps \approx 1 session) Rewards per Bandit Step for Top Performers in Group B (>2)

The data gathered from Group B is not sufficient to make a strong point on bandit learning from the smaller space exploration, even though we see an overall tendency of growth. More data would be required to justify the bandit algorithm learning for this group.

Tutor A shows, though, promising results for the trainees who followed the training program for more than 12 sessions. The average values per step and the moving average show an overall increase and, to understand if there is any learning from the bandit, we will take a closer look at the average values and also the values per each top athlete in Group A.

We assessed the occurrence of high rewards (values greater than 0.5) over each step, and we computed the moving average for this array. The results can be seen in Figure 60. In the first 50 bandit steps, the maximum reached was 3 users with a high reward, the value was obtained in three bandit steps (47, 48, and 49). For the second window, 50-100 bandit steps, the values are greater overall at most of the steps, and the maximum reached was 5 users with high reward, which was in 8 bandit steps (69, 70, 71, 72, 73, 74, 83, 85).

For the last window, 100-144, we saw an overall increase in the high reward. The maximum number of users who reached high reward in this window was 6, in 9 bandit steps (111, 112, 121, 130, 131, 132, 133, 134, 135). As the way the reward is computed makes the bandit be agnostic to either an exercise is too easy or too hard (the reward takes into consideration only the absolute value) while comparing the RIR reported by the user per each bandit step we observed there was no learning of

positive or negative effect of an exercise. We observed that in the advanced steps of the training program athletes received high difficulty exercises which were marked with negative RIR. This will be discussed further in Chapter 0, where we will try to determine what are the potential reasons which determined some users to quit, and if there is any link with the lowest negative RIR reported.

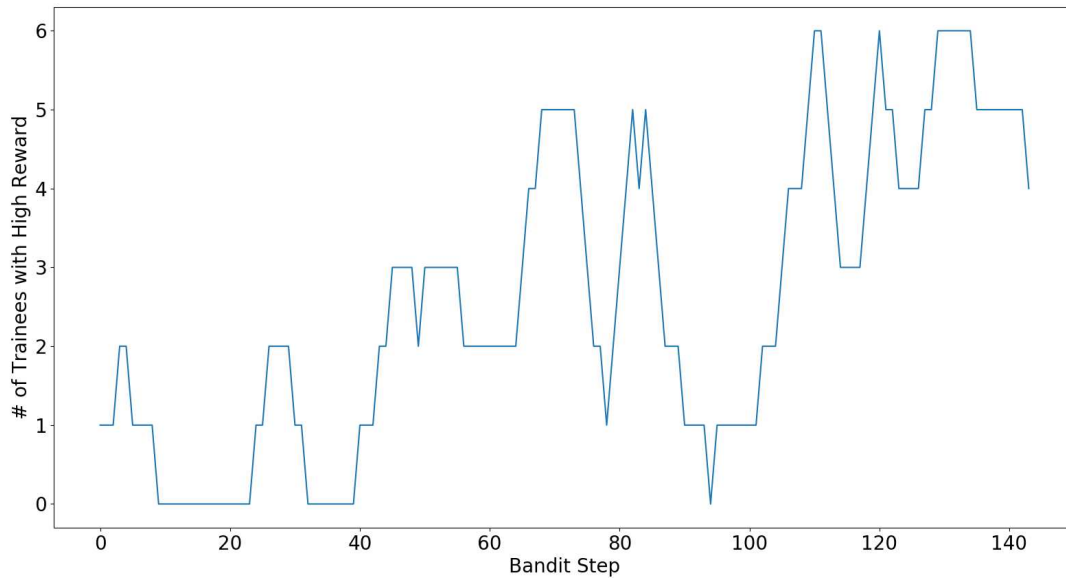


Figure 60. Number of Trainees with High Reward (≥ 0.5) per Bandit Step.

We also plotted the values for each athlete, and we aimed to see if there is any visible increase in the reward. Figure 61 shows the rewards on a moving average of windows size 6 for each trainee of the top 12 performers, at each bandit step. We displayed the prediction steps for each athlete, until step 144 (when at least 3 trainees are following the program). The top 3 trainees are: athlete #1, on the 5th line, athlete #2, the 2nd line, and athlete #3, the 4th line. The smallest number of prediction steps was 57, for athlete #12, on the 8th line.

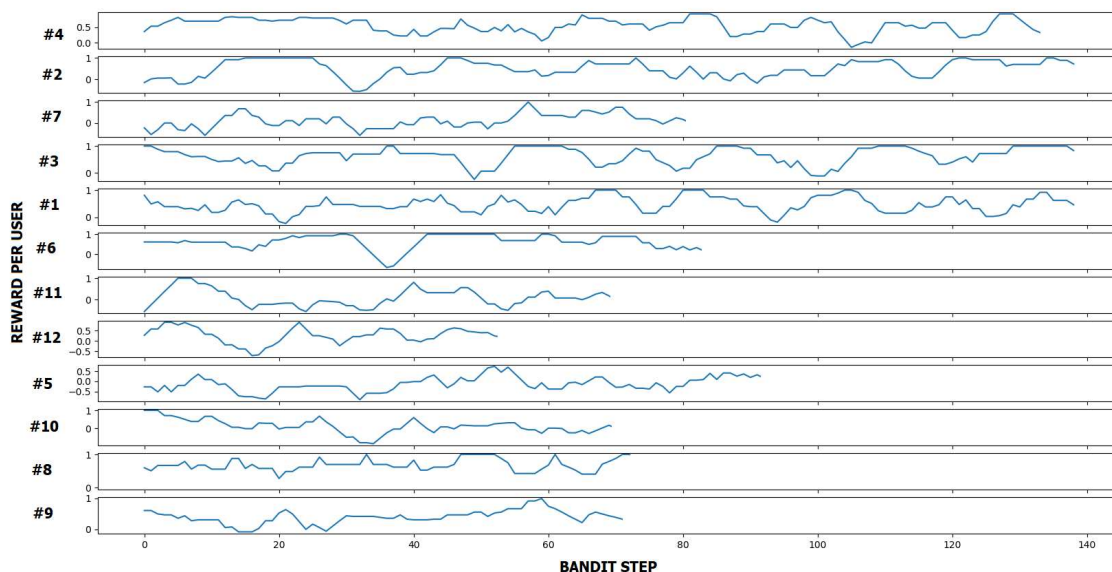


Figure 61. Moving Average (window size = 6 steps \approx 1 session) Rewards per User and for Top Performers per Bandit Step in Group A (\geq 12 sessions)

The reward at each bandit prediction has values between -1 and 1. User IDs from Figure 61 are consistent with the values described in Table 27. Even though for some users it is not obvious from the plots that the overall reward increase over time - e.g., user IDs #6, or #10, there are many users for which we can observe a tendency of growth for the average reward – for e.g., user IDs #3, #2, or #1, which are also the top trainees in the group. Moving averages with a window of 6 data points for each user also confirmed the above statement.

We can observe the overall positive trend for some top trainers, so we will further take a closer look at bandits' predictions for a specific athlete and try to determine if the increasing rewards are linked directly with the learning. We will use as an example User ID #3, for which we see an overall growth tendency of the reward. User ID #3 has performed 23 training sessions, having a count of 144 exercises performed, for which the bandit made recommendations.

At the beginning of each training session, in each of the 23 performed, the athlete reported a good training shape. User filled in a monitoring formular before the session with values for fatigue restoration state, willingness to train, sleep quality, and stress level. Values were on a 10-points scale, where higher values denote that the parameter is not at risk. For example, for fatigue – a higher value means no fatigue, and for willingness to train – higher values means the user is motivated to train, and so on. The average for these values is computed further to generate the current user state, which is classified into three categories: not good (when the average value is less or equal to three), good (when the average value is greater than three and less or equal to seven), very good (when the average value is greater than seven). User #3 reported an average value within 3 and 7 for each session, and so the bandits classified him at each step in a good state. This means only one context state has been used for the current user.

The median of the arms array for User ID #3 is 49. However, the overall split of exercise arms at a bandit step for participants in Group A, based on exercise difficulty, was most of the time equally balanced. The exercises were ranked in the four difficulty categories presented in 10.4.

For three bandit steps of User ID #3, there were 174 available arms, for the other three steps there were 173 available arms, and for one step there were 172 arms. On the lower end, one step had 2 available arms (step 65, see also Figure 62), two steps had 5 available arms, one step had 6 available arms, and another one had 7 available arms. Bandit choices on the 144 steps were the following: 103 steps used arms that were chosen only once (71.5%), 19 arms were chosen twice (26.3%), and only

one arm has been chosen three times (2.2%)— arm/exercise ID 480, which can be also seen in the first line in Figure 62.

Each line in Figure 62 presents a step where the bandit must take an action. The list of available arms is shown on each line, and under each arm, the corresponding existing reward and the current step are shown, "-" value means that the arm was not selected yet. On each line, it is also displayed the selected arm, with a green background, and the reward obtained after executing that action is shown on the same line, on the right, in green. As the exploration factor was set to a low value, 10%, we can observe e.g., for Step 11 the choice made by the bandit looks consistent with the expectations: only arm 480 has been previously explored, and the current reward is 1, so the bandit chooses again arm 480, and the new reward obtained is 1. Same also for Step 65, which is at line four in the figure.

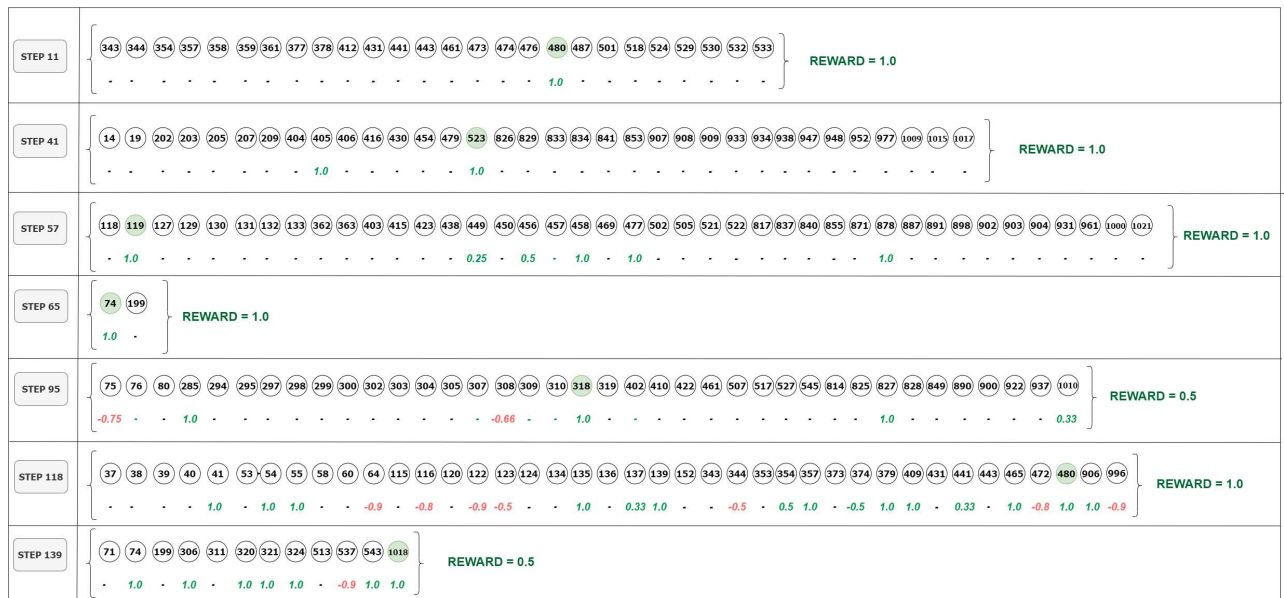


Figure 62. Examples of Bandit Choices for User ID #3 based on the available arms

When there is more than one arm with the highest reward (value 1), the bandit chose one of them randomly, as can be seen in Steps 41, 57, 95, 118, and 139 in the figure. When there was no history in any of the available arms, one value has been picked randomly too. We also observed, while plotting all the data, steps where the bandit chose to explore, even though high or average rewards were already available on some arms. Also, there were cases when the new reward was either positive or negative for existing arms. We can conclude that bandit's choices impacted the overall increase of rewards for top trainers in Group A, and so the bandits' learning is in line with our expectations.

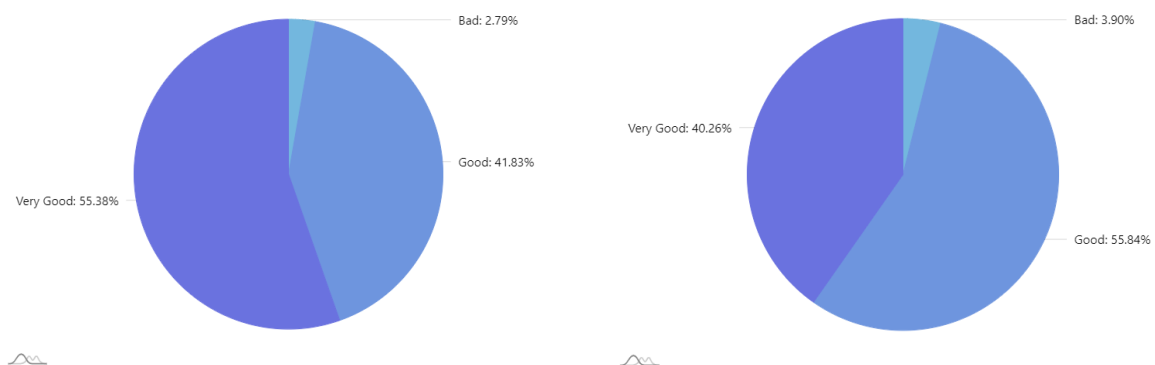
When comparing the number of available arms between the two groups, we observe a notable difference - which we expected - between Group A and Group B. Group A, with a bigger exploration space, has the average value of the available arms array at 31.91, and the median is 19 For Group B, where the exploration space was smaller, the average value of available arms is 10.82,

and the median is 7, which is almost three times smaller than Group A. At a closer look at the number of arms in the arrays, we observed there were 40 exercises that had only one available arm in both groups – which for Group A was 2.9% of the data, while for Group B was 17% of the data. We can conclude that Group B could not use bandit learning in 17% of the cases, as there was only one choice available. Also, for Group A there were 69 bandit steps where the bandit had 2 available arms to choose from (5.3% of the data), while in Group B there were 27 steps with 2 available arms (12% of the data).

Even though the exploration space is narrow for Group B, and the bandit should recommend the exercises close to the user’s level per each movement area, the user states have been diversified across the training sessions. Figure 63 (b) shows that for Group B the split was the following: 3.9% - Bad states, 40.26% Very Good states, and 55.84% Good states. The states were computed as follows - the average of values (motivation to train, fatigue level, sleep quality, and stress level) is received from the user, on a 10-point scale, where lower values mean that the parameter is at risk. States are determined as follows - bad state (average ≤ 3), good state ($3 < \text{average} \leq 7$), and very good state (average ≥ 7). Similar results were also found for Group A, and these can be seen in Figure 63 (a).

Based on the variability of states, certain users require specific exercises. More challenging exercises for a very good training shape or easier ones for a bad shape may prove efficient. Participants in Group B have received recommendations based on their current estimated level per movement area. One potential reason participants in Group B quit faster the training program than participants in Group A was not tailoring the content with higher variability of exercises.

Also, due to the limited exploration space, in some cases there was not much bandit learning and, even though users rated with a negative score or suboptimal an exercise, in future pieces of training the same one was retrieved, as no other choices were available.



(a) Group A

(b) Group B

Figure 63. Bandit Context - User States reported before performing the training session, per each group.

Even though the exercise was marked with a high reward at a certain point, choosing it multiple times may prove boring for some users. The high difference in the number of pieces of training between the two groups was also caused by a group of amateur athletes, from Group A, who trained intensively, and pushed each other to pursue the training program for the experiment.

10.6 Results on User Experience

Another goal of the experiment was to also assess if the *Selfit* system can properly communicate with the learner in open environments. The exchange of information between the tutor and the learner is a critical component in the intelligent tutoring system field (Woolf, 2010).

The current chapter introduces an overarching presentation of the qualitative analysis performed based on user surveys in both versions of *Selfit* – first, at the end of February 2021, and second, at the end of May 2022. The groups involved in each testing phase share similar characteristics (similar age group split, similar split of gender, similar professions), and some users were involved in both experiments. The user experience survey aimed to assess whether the system is pleasant, easy to use, and efficient for the learners.

Multiple iterations were performed, and we adopted an agile development methodology with incremental improvements based on feedback received from trainees. The improvements available in *Selfit version 2* will be discussed and the outcome seen in the differences between the two versions of responses to surveys is presented. Finally, the improvements proposed by the users in the future versions and the overall feedback are discussed.

10.6.1 Results on User Experience – *Selfit version 1* – January-February 2021

Selfit was initially tested with real users at the beginning of 2021, to assess if it meets its requirements (quality test) and monitor whether trainees utilize it with effectiveness, efficiency, and satisfaction (usability test). Due to a few usability bugs, the preliminary testing phase has been stopped and an improved version has been implemented in the months after. Below, we will describe the setting of the initial experiment – *Selfit Version 1*.

A group of 18 trainees, from France and Romania, both novice and experimented in psychomotor training, have used *Selfit Version 1*. The selected population has diversity in training planification and

is matching the simulation strategy: 7 users are under 20 years old, 10 users are between 20 and 40, and 1 user is older than 40.

55.6% of the population is represented by males (10 users – 55.5%), while the others are female (8 users – 44.5%). Each trainee was assigned randomly at registration into one of the training strategies: random, MAB UCB1, ϵ -Greedy (0.1), and Bayesian MAB UCB1. Consequently, they received different recommendations for each session, based on the agent prediction.

Their progress is assessed using sports tests or challenges, but also the subjective feedback got from the users while training: the shape of the day, number of repetitions in reserve per exercise, or post-session feedback.

A user experience survey was conducted during the testing phase with 18 trainees. The user experience (ISO9241-210, 2019) is defined as a person's perceptions and responses that result from the use or anticipated use of *Selfit*. The AttrakDiff questionnaire (Hassenzahl, 2003) on a 7-point Likert scale was used to assess user experience while focusing on both pragmatic and hedonic qualities. The pragmatic quality of an object refers to how users perceive the system's potential "do-goal" in supporting them to achieve a goal (e.g., predictable, confusing, simple, complicated). The hedonist quality refers to how users perceive their "be-goal", namely the *Selfit*'s potential to support pleasure in use and ownership (e.g., boring, interesting, novel, disappointing).

The results for the preliminary *Selfit Version 1* user experience questionnaire are presented in Table 28. Mean and Standard Deviation values are corresponding to qualities scored on a 7-point Likert scale. The mean and standard deviation values are computed for each of the pragmatic and hedonist qualities.

Table 28. *Selfit Version 1* User Experience Feedback based on AttrakDiff questionnaire.

UX Quality	M (SD)	UX Quality	M (SD)	UX Quality	M (SD)
Pleasant	5.44 (1.11)	Connective	4.61 (1.53)	Human	4.55 (1.25)
Inventive	4.94 (1.80)	Simple	4.50 (1.42)	Professional	4.83 (1.50)
Attractive	5.05 (1.22)	Practical	5.50 (0.89)	Likeable	5.83 (0.95)
Straightforward	5.05 (1.17)	Stylish	5.00 (1.20)	Predictable	4.27 (1.19)
Premium	4.66 (1.29)	Integrating	5.72 (0.80)	Brings people closer	4.72 (1.32)
Novel	5.22 (1.35)	Motivating	5.44 (0.95)	Captivating	5.44 (0.89)

The results for the pragmatic qualities show that the *Selfit Version 1* is generally perceived as practical, predictable, simple, connective, and human-oriented. For the hedonist qualities, the system is perceived as stylish, motivating, novel, and captivating. The usability and training generation bugs stopped this phase of testing and an improved version, *Selfit version 2*, has been developed and rolled out to production in the months after.

10.6.2 Results on User Experience – *Selfit Version 2* – January-May 2022

This phase lasted for 5 months, between January and May 2022, and involved 42 trainees, from France and Romania. They got enrolled in the system on different dates and trained between 1 and 30 sessions, the median value was 5 sessions. *Selfit Version 2* did not store any personal data about the user. They enrolled within the system with a nickname, a password, and a security question, as a backup for forgetting the password. No real data about the user was required or either available to fill in, such as email, phone number, name, or address.

After each logging into the system, users could configure their profile, filling in general information on their gender, birth year, bio, or setting up a profile picture. Out of the 42 users, only 10 users filled in the optional information in their profiles. Only the bio and the profile picture fields were public and visible to the other members enrolled in the system. All the users were involved in the pre-test - the calibration challenge performed when they started to train using *Selfit*. 10 of them also completed the post-test, at the end of their training program.

At the end of May 2022, the trainees received a user experience survey to fill in. The structure of the survey was the same as the one described in the previous section, for *Selfit Version 1*, based on the AttrakDiff questionnaire, and included 9 new questions, where we aimed to assess the training shape of the user, involvement in the experiment, and some open questions on the overall perception of the user. 21 trainees out of the initial 42 (50%) filled in the questionnaire. Based on our knowledge,

most of the trainees involved in the *Selfit Version 1* experiment also used the new version and trained from January-May 2022. We aimed to assess if there is any visible improvement in the overall User Experience between the two versions, what were the main pain points and strengths of the system, and if there is any link between the motivation to train and user experience.

The AttrakDiff questionnaire values for *Selfit Version 2* can be seen in Table 29. Values in bold mark a difference greater than 0.5 on the 7-point Likert scale between the two evaluations. We can state that overall user experience has improved in *Selfit Version 2*, as many hedonists and pragmatic quality values are better. The negative differences between the two versions indicate that the users perceive the new system as a bit more technical overall and feel like the system does not fulfill their need on bringing people together, the social component.

Table 29. *Selfit Version 2* User Experience Feedback based on AttrakDiff questionnaire.

UX Quality	M (SD)	UX Quality	M (SD)	UX Quality	M (SD)
Pleasant	5.90 (1.26)	Connective	4.14 (1.64)	Human	3.85 (1.38)
Inventive	5.76 (1.71)	Simple	5.66 (1.65)	Professional	6.0 (0.83)
Attractive	5.19 (1.56)	Practical	5.66 (1.42)	Likeable	5.95 (1.21)
Straightforward	5.52 (1.36)	Stylish	4.90 (1.44)	Predictable	4.66 (1.49)
Premium	4.76 (1.54)	Integrating	4.95 (1.29)	Brings people closer	4.05 (1.30)
Novel	5.47 (1.49)	Motivating	5.90 (0.74)	Captivating	5.40 (1.39)

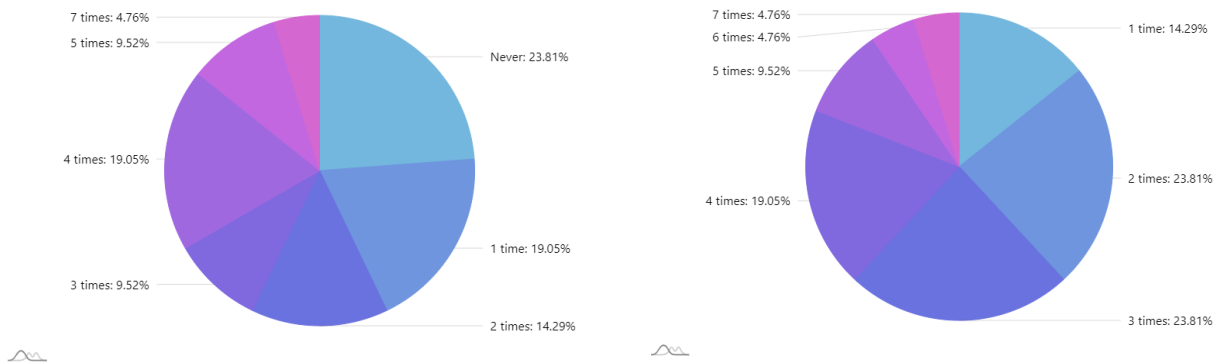
Users involved in the experience survey were asked also to fill in how many trainings they performed on average by the week before and during the experiment. The results can be seen in Figure 64. 5 users involved in testing *Selfit Version 2* were not training at all before the experiment (23.81%), and, e.g., 1 user was training every day of the week (4.76%) – see Figure 64 (a). During the experiment, most users were training either 2 or 3 times per week, 5 users in each category, 3 users trained only one time per week, while the same user as before the experiment continued to train 7 times per week – see Figure 64 (b).

Of the users who did not train at all before the experiment, 2 of them used the system and trained 3 times per week, while the remaining 3 trained once per week. 20 trainees out of the 21 who filled in the survey, either maintained or increased the number of sessions they train per week.

Only one trainee reported that during the experiment he trained twice per week, and before the experiment, he trained four times per week. 6 users reported having doubled the number of sessions

per week during the experiment. Also, 6 users maintained the same number of sessions they trained also before the *Selfit Version 2* testing phase (1 user twice a week, another user 3 times a week, 2 users 4 times a week, 1 user 5 times a week, and 1 user 7 times a week).

Out of the 21 users who filled in the survey, 16 were male (76.2%), and 5 were female (23.8%). As per age category - 19 users are between 20-40 years old, one user was under 20, and one user was over 40. The user profiles were mainly not sportive. 20 users out of the 21 filled in their profession, from which 12 reported they are software developers, 4 were students, 1 was an interior designer, 1 university teaching assistant, 1 researcher, and 1 person working in economics. 11 users filled in the survey also their corresponding nicknames used in the *Selfit Version 2*, to be identified with their corresponding survey data. 7 users were from Group A, while the remaining 4 were from Group B. They performed between 2 and 27 sessions, the mean was 9.63 and the standard deviation was 7.03.



(a) Before the experiment

(b) During the experiment

Figure 64. Number of training sessions per week reported in the survey

Users had to rank also in the survey the top features, up to 7 from a list of 17 features most asked for while training, features they want to see in the next version of *Selfit*. The most voted features can be seen in Table 30. Three features were voted 13 times each by the users, most votes obtained per feature (61.9% of the total amount of users). Most users voted for "voice or sound guidance while training", which means, for the time-based exercises, the last seconds on the chronometer to also be announced with a sound, as the users do not follow all the time their mobile phone screens. Some users also proposed for this feature to be able to move within the app (next, pause, resume, previous buttons) using voice commands.

Table 30. Top ranked features proposed by the users for the next version of *Selfit*.

Feature description	Number of votes	% of users who want the feature
Voice or sound guidance while training	13	61.9 %
Auto Login / Remember me	13	61.9 %
Tips for each exercise / Extra description to understand better	13	61.9 %
Fix all the missing videos, improve some existing videos	12	57.1 %
Stretching module after training / Add Pilates, yoga exercises	10	47.6 %
Display more performance graphs / More statistics	10	47.6 %
Make warm-ups shorter, focus more on the core exercises	9	42.9 %

Another top-voted feature was "tips for each exercise / extra description to understand better". This feature was reported also during the experiment, users have reported that the video for some exercises is not enough to perform the exercise and a full description of body position and movements involved would be required. As for the third top-voted feature, "auto-login / remember me", we can state that this is more a missing implementation than a feature. Users reported that inputting the username and password each time they enter the application may become frustrating, and we agree this functionality should be put in place in the next *Selfit* version. This behavior was only reproducible for Android-based devices.

On the lower end of the voted features, 1 user asked for a way to fully customize his session and be able to select all the exercises to train from a list. 2 users asked for a smarter way to input RIR, and not fill in many inputs per each exercise. It is possible that the voice integration feature, which is a way to communicate with the tutor only by speaking (Natural Language Tutors), will help fix this issue. But this will be discussed further in the next chapter. 5 users also asked for a picture or a teaser of the exercises when they are presented on the summary page, to know what to expect.

The survey included also two open questions: one to fill in if there is any other feature, different from the ones in the list, which they would like to see, and second – if there is any overall feedback they want to provide. Both questions were optional. For the first one, 7 answers have been received, while for the second – 11 answers. One user proposed for the future to provide tips related to nutrition and how and when to drink water while training. Another one proposed to pre-store the videos for exercises and through this, speed up the loading time for each exercise page.

The overall feedback while using the system was also assessed. Most of the feedback was positive and constructive, users said that *Selfit* is challenging and fun, and they perceive it as a proper system for a healthier lifestyle. The word cloud of the overall feedback in the user experience survey can be seen in Figure 65. Two users said they were happy to be able to participate in this experiment.



Figure 65. Word Cloud with Overall Feedback from User Experience Survey

One user was even more enthusiastic, his last words on the survey were: "The app is innovative, I started going to the gym because of it, so I consider it a great success". Two users said the exercises' description and some missing videos make the system not fully ready, but they would expect to integrate it into their daily pieces of training if these features and missing functions are implemented. It worth mentioning though that we do not know what is the percentage of participants from each group who fills in the survey. We can estimate, based on their response to the questions (e.g. number of times trained during the experiment), that the majority of users filling in the survey was from Group A. We also matched the ones who filled in the username question - 63% (7 users) were from Group A, while 37% (4 users from Group B).

Discussion and Conclusions

11 Discussion and Conclusions

11.1 Résumé

Ce chapitre met en évidence les avantages de l'approche proposée et les défis rencontrés au cours du processus de développement. Chacun des objectifs de recherche initialement abordés est maintenant discuté, sur la base des propositions présentées dans le chapitre précédent. Tout d'abord, pour construire un modèle de représentation des connaissances efficace pour le développement des compétences psychomotrices dans un STI, nous avons introduit OntoStrength, qui a ensuite été intégré dans le système Selfit. L'ontologie tient compte de la diversité des activités de mouvement humain et de leur description, du profil métabolique du mouvement et des règles et contraintes de développement spécifiques des différentes sous-compétences utilisées par le programme de développement. L'évaluation des compétences psychomotrices est également prise en compte dans l'ontologie.

L'approche de la personnalisation de l'apprentissage des compétences psychomotrices dans cette étude a utilisé l'ontologie OntoStrength pour recueillir des informations sur les caractéristiques du stagiaire, telles que les traits morphologiques et physiologiques, les antécédents de blessures et les niveaux de forme physique quotidiens. Le contexte du stagiaire et l'environnement d'entraînement disponible ont également été pris en compte dans la personnalisation de l'apprentissage. L'algorithme Contextual Multi-Armed Bandits (Contextual MaB) a été utilisé pour décider des meilleurs exercices d'entraînement en fonction du contexte du stagiaire. Les résultats d'une expérience menée auprès de 42 utilisateurs ont montré que l'espace d'exploration avait un impact significatif sur la motivation à s'entraîner et qu'un large espace d'exploration contribuait à maintenir la motivation. Les meilleurs stagiaires ont vu leur récompense moyenne augmenter au fil du temps, ce qui démontre l'efficacité de l'approche de l'apprentissage personnalisé.

Le module de communication est un élément essentiel du système Selfit, un système de tutorat intelligent (STI) conçu pour soutenir le développement des compétences psychomotrices. Il facilite l'échange d'informations entre les utilisateurs et le tuteur, ce qui permet d'évaluer efficacement les progrès du stagiaire sportif. Le module de communication comporte plusieurs sous-modules, dont l'authentification, l'étalonnage du stagiaire, le retour d'information et l'entraînement. Le module de communication a été testé avec des utilisateurs réels et les réactions ont été positives, les utilisateurs percevant Selfit comme agréable, inventif, simple et professionnel.

Ce travail présente plusieurs limites, notamment la taille limitée de l'échantillon, un degré limité de généralisation en raison d'un faible budget de promotion et du fait que les participants étaient principalement de jeunes hommes travaillant dans le développement de logiciels, les limites de la collecte de données et de la mesure de l'effort perçu, la complexité du domaine psychomoteur de la force, une base de code importante avec un indice de maintenabilité moyen et l'absence de tests exhaustifs. La thèse a montré que des efforts supplémentaires sont nécessaires en termes d'augmentation de la taille de l'échantillon, de meilleure promotion de l'expérience auprès de diverses populations, de méthodes de collecte de données plus précises et de tests complets de la base de code. *OntoStrength* contient une description complète du profil psychomoteur et des programmes d'entraînement et sera publié gratuitement sous une licence ouverte.

11.2 Introduction

The last part of the thesis discusses the empirical studies and based on the previous findings, we address the research questions raised in the Introduction. This part starts with the Discussion chapter, which describes the advantages of the approach, the limitations, and the envisioned applications. The Discussion chapter summarizes the methods and systems proposed, aligned with the three research questions, on different pillars of intelligent tutoring systems modeling for psychomotor training in open environments. First, we present the advantages of our approach to the knowledge modeling pillar. Then, we summarize the approach to psychomotor tutoring, and lastly, we discuss the communication modeling by introducing the *Selfit* system.

A set of known limitations are introduced further, split into three categories. The conducted experiment had a limitation in terms of sample size. The participants involved were not matching the initially performed simulations and, for the ones who got involved, many quit before the experiment ended, so only partial data was available. Another limitation that will be elaborated on further is the degree of generalizability. This was caused by the method we recruited the participants for the experiment. We promoted the experiment within our social circles, which got in the experiment similar trainee profiles. Another limitation was the lack of resources which did not allow us to perform a better assessment and will be described in the Limitations subchapter.

The last section of the Discussions chapter presents a list of Envisioned Applications, where each of the developed systems can be applied in future psychomotor research projects. *OntoStrength*, on one hand, can be used as a knowledgebase for other sports training scenarios, and be extended to other fields, which are briefly discussed further. *RiARiT* method can be applied in other learning scenarios, and the modules created within the *Selfit* system, which includes assessment modules or

user interface components, via phone sensors, cameras, or integrations with third-party wearables suppliers or computer vision libraries can be used further in other educational contexts.

This part ends with the Conclusions chapter, which was split into two sections – personal contributions, and a list of directions for future work. The personal contributions section introduces the timeline of work for the current thesis, with important milestones, conferences, or journals where methods or systems have been published, and also the researchers who were involved in each sub-project. We estimate the current thesis may have a high impact on improving the general health of the mass population, so we conclude our work by drawing potential directions for future work, out of which our team already started exploring some.

12 Discussion

Based on the findings of the literature review performed initially, we have designed and developed an Intelligent Tutoring System for psychomotor training. Without limiting the overall perspective, we introduce for the current thesis and focus solely on *Selfit*, an ITS used for psychomotor training in open environments, and specialized in strength training.

We addressed the challenges raised and evaluate the system in terms of personalized learning efficacy, first within simulations, and then with real users who used two adaptive strategies versions of the *Selfit* system. Next, we will present the advantages of our approach, mapping the challenges raised initially. Afterward, we will outline the limitations of the study and we will finish by presenting the educational implications within a few envisioned scenarios.

12.1 Advantages of our Approach

The thesis goal is to provide a unified approach that tackles challenges discovered when building an Intelligent Tutoring System for psychomotor training. Even though there are hundreds of mobile or web applications available on the internet which act as virtual sports coaches, for most of them we could not find the scientific grounding in the work. This may be also justified by the business model of these systems, which does not allow sharing of information publicly.

Also, from the testing performed by our team on some of the most known fitness training applications, we could not assess the personalization features, which implies training content adaptation based on different trainees' profiles. The overall perception of the recommended training content in the most popular physical training apps was that they rely on fixed rules and there is no adaptive tutoring.

As we could not link any literature research on sports training with existing digital sport coaches available on the market, we can consider the current work can open new research paths at the crossroad of artificial intelligence, human-computer interaction, and sports training. In order to highlight the benefits of our approaches, we will present the methods and findings for each research objective below.

12.1.1 Knowledge Modeling in Psychomotor Training

RO1: Design an effective knowledge representation model for the psychomotor skills development in an Intelligent Tutoring System.

The effort of building psychomotor development knowledge requires multidisciplinary expertise. Our review of the literature did not find any previous work to map the description of the psychomotor field. Existing work found focuses on recognizing sports activities from videos, or supporting decision-making based on the data collected during sports competitions. The intelligent tutoring systems for psychomotor skills developed so far are not many. The proposed methods and findings address various sub-fields, including learning to drive a car, gaining skills for laparoscopic surgeries based on robotics, postural rehabilitation for health, military training, and ball-passing training. Except for the GIFT project, developed for military training (R. A. Sottolare, Brawner, & Sinatra, 2017), the other existing work in psychomotor ITSs does not clearly outline how the knowledge is modeled. Cross-disciplinary teams of the GIFT NATO department developed and described the knowledge modeling for training marksmanship at the GIFT Symposium every year since 2016²¹, proposing also recently ontology-driven approaches for the modeling (Brawner et al., 2019).

The current thesis introduces an ontology to model learning in the psychomotor field, called *OntoStrength*. The proposed modeling is dedicated to enhancing psychomotor skills and associated bio-motor abilities, with a more comprehensive description of the strength skill. *OntoStrength* has been built by our team with cross-disciplinary expertise, with a background in muscular, biomechanical, sports training, and computer science fields. *OntoStrength* models knowledge for training development program periodization, which includes Macro Cycle, Meso Cycle, Micro-Cycle, Workout, Warm-Up, Content, Cool Down, and Exercise classes.

The modeling presented in this thesis describes the knowledge of the trainee's psychomotor capacities, e.g. Muscle Contraction, Fundamental Movement, Hip Movement, Ankle Movement, or Movement Skill. *OntoStrength* models a learner individualization signature, which is specialized in bio-motor skills: Flexibility, Endurance, Speed, Strength, and Injury. It was developed using Ontology 101 Methodology (OD101), SPARQL was used for queries, and the views were implemented using the semantic graph database GraphDB. OWL is the formal representation language, and the software used for development was Protégé.

OntoStrength was further integrated and used within an intelligent tutoring system for psychomotor training, called *Selfit*. The knowledge described was mapped to the ITS components. Domain module addresses the movement field by first considering the diversity of human movement activities – daily life, leisure, or professional – and their description – muscular contraction, human body joints

²¹ <https://gifttutoring.org/projects/gift/wiki/Overview>

movements, or fundamental movements. Second, it includes the movement metabolic profile with its duration and intensity. Moreover, the domain is also related to the specific development rules and constraints of the different sub-skills used by the development program.

The student module considers the evaluation domain of psychomotor skills. From this perspective, *OntoStrength* includes performance indicators and associated evaluation rules for each sub-skill considered by the development program. The ontology also addresses performance indicators used to monitor trainee responses to the training workout and to adjust the planned program to the reality of the effects of its application.

The development of *OntoStrength* – our proposal for knowledge modeling in the psychomotor field – was challenging because of the heterogeneity of domains to be integrated, each with its independent design. Moreover, different cultures and approaches co-exist in each domain, generating the necessity to organize concepts sometimes as complementary and sometimes as antagonists.

The ontology was mainly developed based on the sports training theory described by Bompa (2017) and recent work on the psychomotor field, including the updates of the psychomotor taxonomy (Hoque, 2016), and similar work developed for other use cases: Goldberg et al. (2018) – for military training, or PRB de Campos (2018)- for learning engineering. The rise of the Semantic Web, the recent advancements in the digitization of the psychomotor training, and our team's expertise in the field were the main facilitators for the proposed ontology. Nevertheless, *OntoStrength* has a central role in the current thesis as it grounded the development of effective tutoring and the elements rendered in the graphical user interface.

12.1.2 Teaching Strategies in Psychomotor Training

RO2: Provide personalized sports exercise recommendations for the mass population when training in open environments.

When designing an ITS, one of the main features to enable is personalized learning, which is the tutor's capability to generate and adjust development tasks according to trainee characteristics. The psychomotor field has a set of characteristics that should be at the foundation of training planification. The first one is gender. The morphological, cognitive, and physiological differences between males and females impact the content development of the training sessions, together with associated risks of injuries and psychological disorders. The second set of characteristics relates to trainee level on different performance skills. Some learning tasks have specific performance prerequisites to be enabled before being provided to the trainees. The third set of characteristics

reflects the training environment, which includes the following variables: training location, available time, and materials to train.

Nevertheless, numerous external factors might influence trainee fitness, fatigue, and willingness to train, such as daily life, professional, or other psychomotor activities. External factors which impact the training content are the fourth set of characteristics. The consequence is that trainee fitness and fatigue levels might be superior or inferior to the state assumed by the tutoring system. In addition, when starting a new training session users might be injured and incapable to perform part of the psychomotor training tasks.

The challenge in psychomotor training is determining the most effective sequence of activities that will maximize the average competence level across all skills for each individual trainee. This is difficult to achieve because trainees have a limited amount of time to practice and cannot test all available exercises to determine which are the most effective for them. Additionally, there are individual differences among trainees, so a task that is optimal for one person may not be effective for someone else with a similar profile.

Our approach to personalizing learning for psychomotor skills used *OntoStrength* ontology, which provided the required information to feed the four sets of characteristics described above. *OntoStrength* provides detailed information on psychomotor learning components. Each exercise is described by the required materials, set of psychomotor movements implied, muscles involved, number of joints, and difficulty (labeled between 1 – very easy and 10 – very hard). *OntoStrength* maps a comprehensive trainee profile, with morphological, and physiological characteristics, a history of injuries, diseases, and daily reported shape (sleep quality, fatigue, motivation to train). The first phase in personalizing learning implied filtering based on the characteristics – trainee profile, trainee current state, and training environment.

Next, we used Contextual Multi-armed Bandits (also known as Contextual MaB) to enhance the personalized learning experience, as can be seen in Figure 66. Even though there are many variables that generate the current choices/arms, these are encapsulated in the filtering logic applied before the bandits, which receive only the trainee identifier, the trainee context, and a list of arms. A similar approach was recently implemented in the cognitive field, for an ITS which teaches basic mathematics operations to 7-8 year-old schoolchildren (Clement et al., 2015), and has shown promising results. Multi-Armed Bandits used in the cognitive field proposed even better learning sequences than expert teachers. The Contextual MaB takes a decision based on a given context, in a sequence of independent trials. The goal is to maximize the total pay-off or the reward of the chosen actions.

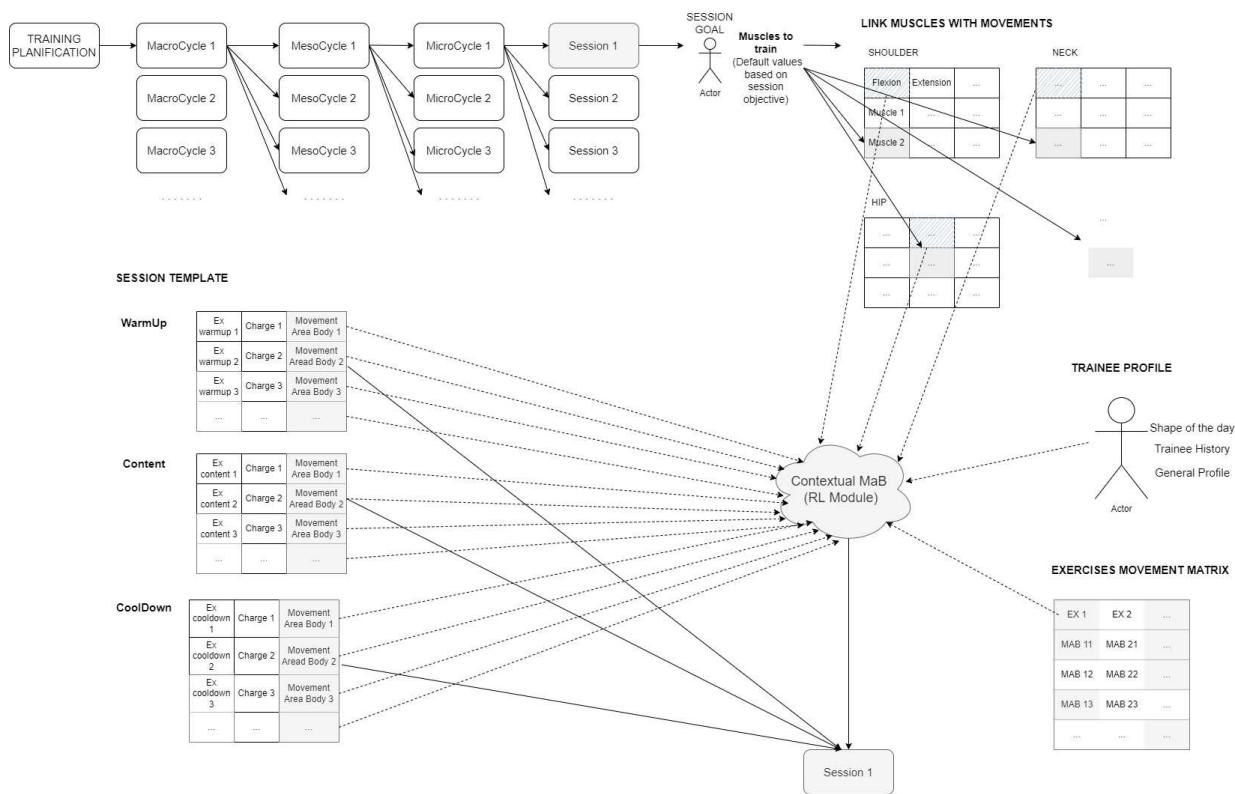


Figure 66. Personalized Learning for Psychomotor Training using Contextual Multi-Armed Bandits.

Our initial implementation in a simulated environment shown this personalizing approach is efficient, and it provides high-performing learning sequences after six months of continuous training. We assessed the effect of Contextual MaB in an experiment with real users, which involved 42 users who trained for 6 months. The participants used one of two versions of the *Selfit* system, with two adaptive strategies – Tutor A with a wider exploration space (simple and difficult exercises shown to the users in Group A), and Tutor B, with a narrow exploration space (only current user level exercises shown to the users in Group B). The participants were asked to perform 12 training sessions, and also an initial and final calibration session.

We observed the participants in Group B have quit much faster the trainings – no user performed the full training program (11 training sessions for the top trainer in this group), while in Group A there were 12 participants who followed the full training program, and also some top trainees – the top three performed 30, 27, and 25 sessions. This was a surprise for us, as the initial experiment setup assumed similar engagement overall, and even Group B to surpass Group A training time.

The variability of the trainee state (low/high fatigue, or low/high motivation) and bandit' learning over time impacted the choice of exercises when generating a training session. Group A has demonstrated a higher motivation to train overall, and this can be justified by the wide exploration space, which helped the bandit tailor the session with easier or harder exercises for each trainee current context. We can conclude there was a high impact of the exploration space parameter (*Selfit*

Tutor A vs. *Selfit Tutor B*) in the choice of exercises and finally seen in the overall motivation to train due to user's sensitivity to non-average exercises, based on his state. Our findings highlight the importance of the diversity of exercises, through wide exploration, in order to maintain the motivation to train and keep the user engaged.

The results for top trainees showed an increase in the average reward across time and, even though they did not follow the exact training planification as for the simulation, the results of the experiments are slowly converging to the simulations, which was in line with our expectations. The Contextual MaB method for personalizing learning in psychomotor training has shown promising results overall.

12.1.3 Interaction with the Trainee in Psychomotor Training

RO3: Implement an intuitive and effective Communication Module that facilitates the assessment of the sport trainee's progress in open environments.

It takes a lot of effort to create the communication module of an Intelligent Tutoring System (ITS). Some aspects, such as creating interfaces and using animated agents, are relatively easy, but developing natural language systems can be more challenging. Good communication skills are important for individuals who work with others, especially for teachers (Woolf, 2010).

Current work introduces a Communication module to support the exchange of information between the users and the tutor, to support psychomotor skills development. The Communication module interacts with the other three components on the ITS – domain, student, and tutoring modules. The Communication module was integrated within the *Selfit* system, and it uses OntoStrength knowledge modeling and the Contextual Multi-armed Bandits method for personalized learning.

Feedback is an essential dimension of psychomotor development and consequently to *Selfit* efficiency. The trainees using *Selfit* access the interface and communicates with the tutor through a Progressive Web App (PWA), available on cell phones, tablets, or computers, via a web browser, or on the most known mobile stores – Android - Google Play Store²² and Apple – App Store²³. The development has been performed on the Microsoft ASP.NET platform – ASP .NET Core, using Razor syntax, and Entity Framework.

²² <https://play.google.com/store>

²³ <https://www.apple.com/app-store>

The Communication module proposed has the following components: (a) Authentication component, (b) Trainee Calibration component, (c) Feedback component (self-evaluation – before and after session, exercise-level feedback, per each set – acute feedback), and (d) Training workout component (video demonstrating the movement, workout setup, workout summary, past workouts). Trainees are required to create an account at their first interaction with the *Selffit* system. No personal data is stored in their user profile. They are asked to log in and input their nickname and password, or login with the security question each time they use a new device or after the session token expires for that device.

Trainee Calibration sub-module is a set of component interfaces that has the role to exchange information with the user to estimate his current level across the main movement areas – upper body (push horizontal, push vertical, pull horizontal, pull vertical), lower body (hip dominant, knee dominant). The Calibration module includes a maximum of 24 exercises to execute, 4 per each movement area, ranked based on the difficulty (from level 1 to level 4). If a specific level is not passed, the more difficult exercises of that category – higher levels – will not be shown. To pass a specific level, the user has to watch the video with those specific exercises, execute the number of repetitions shown, and then input the number of executions he was able to execute. The future sessions will be tailored based on the feedback reported per each movement category. Trainee Calibration is required before the first session using *Selffit*, and it will be available at any time for users on the home page. It is recommended to redo the calibration challenge after finishing a mesocycle and to be in a good shape when performing it.

The feedback sub-module is critical for personalizing the learning. The communication module assesses the user state before starting a session (to determine user context), at the end of the session (to determine the rate of perceived effort), and inside a session, per each exercise (to assess the number of repetitions in reserve). Based on our findings, the existing applications in sports training also assess the initial and end state, but not the granular feedback per exercise, which is of high interest in psychomotor personalization.

Training workout component exchanges information with the learner on the training program and includes the setup of the training (location to train, time to train, available materials, preferred muscles to target), the training summary page, the past pieces of training list, and the flow of training. While training, a video demonstrating the movement and details required to accomplish the correct load (i.e., number of sets, number of repetitions, rest between repetitions and between sets) is displayed for each exercise.

The Communication module of the Selfit system received positive feedback from real users who participated in a user experience survey using the AttrakDiff questionnaire. Out of the 21 users who completed the questionnaire, the overall consensus was that *Selfit* was perceived as pleasant, innovative, straightforward, and professional.

12.2 Limitations

Our work has a set of limitations that are important to highlight, not only for the current research process but also for future researchers to consider in framing their studies. Next, we will outline the known limitations and a description of each of them.

12.2.1 Sample Size

The dataset obtained during data collection is not fully matching the size of the simulations. Simulations performed in the virtual environment have demonstrated the learning of the contextual multi-armed bandits' algorithms. These showed that the users should train for at least 48 sessions to have a visible increase in the average reward across sessions, while in the real experiment the top performer trained 30 times. Due to the limited size of participants and time availability, we did not test all the three bandits' algorithms used in simulations. Even though for this experiment we could not create a larger dataset, the results obtained shown the values are converging to the simulated data. We also assessed several decisions taken by the bandits in different steps, and we have proven that it is learning.

12.2.2 Degree of Generalizability

Even though we published the system on the mobile stores and the Internet, and made it available across the world, for free, we had a limited budget to promote it and to reach a high number of people. We promoted the experiment on our social platforms (LinkedIn, Instagram, and Facebook), and we gathered 42 trainees who enrolled in the program. Most of them were beginners in the sports training field, as shown in the initial calibration challenge. Very few of them were intermediate or advanced, being involved in other sports activities too. Their professions were mainly software developers, then scientific researchers, teaching assistants, or students. The majority was composed of males. Also, more than 75% of the users had between 20-40 years old. The sports individualization theory applies the same rules based on age differences and professions, considering the description of the trainee profile – gender, injuries profile, diseases. Our findings support the generalizability of the system up to a certain degree. Younger (<20 years old) and older (>40 years old) are exposed to

different injuries and diseases than the existing population. Our findings can act as a springboard for future research to look at different populations.

12.2.3 Resource

In terms of data collection while training, the measurement of perceived effort per each exercise – repetitions in reserve (RIR) has been assessed using the subjective input from the trainee, which can be noisy. A more accurate measurement of the perceived effort is monitoring the heart rate while training and correlating with the input of RIR. The heart rate variability indicates if the user is training or not, and what is the approximate fatigue. It may also be a good indicator of counting the repetitions executed. Measuring the heart rate requires an external smart device, such as a smart band, smartwatch, or chest strap, which the users involved in the experiment required while training. Also, integration with the 3rd party providers of such devices – Google, Apple, Fitbit, Garmin, Polar, and others. We implemented the integration with Garmin and Fitbit providers for heart rate monitoring, but the users involved in the experiment did not have devices provided by these manufacturers to track data while training. We did not have the budget to provide such devices to the population in the experiment.

12.2.4 Complexity of the Strength Psychomotor Domain

Another limitation we mention is the complexity of the strength psychomotor training field, with multiple layers of knowledge representation, inner connections, and inter-connections with other fields. Even though we can conclude we obtained good modeling of the strength field, through *OntoStrength*, which was emphasized by our results, we did not map all the parameters which influence the sports training individualization. This was due to both time constraints and limited knowledge of cross-disciplinary expertise, such as the dietary dimension, where we lacked the required knowledge.

12.2.5 Large Codebase

Also, another limitation is generated by the overarching development of the *Selfit* system, over the last 4 years, which was not heavily tested and may throw bugs in some edge cases. Our team did not have a Quality Assurance (QA) engineer assigned to develop test cases and execute them periodically. This exposes a vulnerability of the *Selfit* system overall. An overview of all available flows and edge cases should be described for any software, to have a better view of the potential bugs which may

occur. Based on the Visual Studio²⁴ analytics using Code Metrics Calculator, the current version of the *Selfit* system has 221.873 lines of code. The analytics show a Code Maintainability Index²⁵ of 74%, which is an average value for a code easy to support and change. More effort should be allocated further to writing test cases and code refactoring for better maintenance in the long term.

12.3 Envisioned Applications

The first part of the thesis has a more theoretical focus and aims to establish the foundation for using intelligent tutoring systems for psychomotor training. Knowledge modeling for this field was afterward introduced. The proposed ontology, called OntoStrength, is the first work, based on our literature findings, to describe the strength psychomotor training. It includes sports movements and movement patterns. OntoStrength has also a comprehensive description of the psychomotor profile and planification for training programs, for each phase – session, micro-cycle, mesocycle, and macro-cycle. The OntoStrength database contains over 1,000 exercises that are labeled based on the muscles involved, movements, joints, materials required, videos, and difficulty. It will be released under an open license and the knowledge base will be made freely available.

The work presented for OntoStrength can be used to improve certain learning scenarios. Other systems targeting strength training can reuse the OntoStrength modeling and extend the work to other psychomotor fields – such as endurance, or flexibility.

The reinforcement learning implementation using contextual multi-armed bandits' algorithms was shown efficient for adaptive personalization of learning activities sequences in the cognitive field (Clement et al., 2015). The approach was proven suitable for scenarios when users have limited time and they might face motivational issues when they are involved in long-term learning. The current thesis used this approach for psychomotor training, also integrating the user's current state (trainee context), and it has shown promising results. The learning activities provided tend to maximize the student's skills over time.

Our approach introduced a Python²⁶-based algorithm called RiERiT (Right Exercise at the Right Time) which estimates the learning progress of the trainee and determines the appropriate exercise to be performed at a given time. This work can be used in other adaptive tutoring scenarios. Other

²⁴ <https://visualstudio.microsoft.com>

²⁵ <https://docs.microsoft.com/en-us/visualstudio/code-quality/code-metrics-maintainability-index-range-and-meaning>

²⁶ <https://www.python.org>

systems which involve an adaptive component, in psychomotor training, may use the current approach as a starting point.

The user interface model has been built as a monolithic system, using Microsoft tech stack – ASP.NET MVC²⁷. The code source will soon be published publicly on the GitLab platform²⁸. Specific features, such as calibration challenges, wearables integration with the most known providers (Garmin²⁹, Fitbit³⁰), session flow, and profile configuration by the users may be reused by other researchers in their systems.

Overall, as governments are more and more interested in mass population health, and the lack of physical activity is a high concern for the more-developed countries, we envision the current work as the foundation of a larger-scale research project, over the next years, with focus to digitize the sports training field and develop several solutions which aim to improve the lifestyle of both beginners and intermediate trainees. We consider the timing is right for such a research project, the technological advancements support the required developments, and our core team can provide the expertise for leading the development of relevant solutions further.

²⁷ <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc>

²⁸ <https://gitlab.com>

²⁹ <https://www.garmin.com/en-US/>

³⁰ <https://www.fitbit.com/>

13 Conclusion

13.1 Résumé

La thèse visait à étudier le rôle des systèmes de tutorat intelligents (STI) dans la formation des compétences psychomotrices et la manière dont ils peuvent être construits et utilisés à cette fin. L'approche comprend une ontologie dans le domaine de la psychomotricité appelée OntoStrength, qui a été utilisée pour la modélisation des connaissances, et une approche de personnalisation basée sur l'apprentissage par renforcement et les bandits multi-bras contextuels. La thèse a apporté des contributions dans de nombreux domaines, notamment l'informatique, la modélisation des connaissances, la science des données, les sciences de l'éducation et les sciences du sport.

Les travaux présentés ont porté sur le développement de trois systèmes : OntoStrength, RiARiT et Selfit. OntoStrength est une ontologie développée par l'auteur avec l'aide de spécialistes du sport et qui a été publiée dans diverses revues. RiARiT est une méthode de personnalisation des séquences d'apprentissage en psychomotricité, qui a été simulée et intégrée dans la composante tutorat d'un système de développement psychomoteur. Selfit est un prototype de STI qui combine les deux contributions précédentes et qui a été testé avec des utilisateurs réels.

Le présent chapitre se termine par des orientations pour les travaux futurs. La méthode actuelle d'évaluation de l'utilisateur dans la thèse repose sur la saisie manuelle de l'effort perçu par l'utilisateur et le nombre de répétitions en réserve. D'autres améliorations peuvent être apportées en intégrant les données de fréquence cardiaque provenant de dispositifs portables, tels que Garmin et Fitbit, et en les utilisant comme indicateur plus précis de l'effort. En outre, la vision par ordinateur peut être intégrée pour mesurer le nombre de répétitions correctes et lancer automatiquement le chronomètre du temps de repos. La personnalisation peut être étendue des novices aux entraîneurs intermédiaires et avancés, ce qui permet une expérience d'entraînement entièrement personnalisable. Le système peut également être étendu pour prendre en compte l'alimentation et la nutrition, et de nouvelles orientations d'entraînement telles que la flexibilité et l'endurance sont en cours de développement.

Insufficient physical activity is a major health concern worldwide, affecting a large number of people in both developed and developing countries (Guthold, Stevens, Riley, & Bull, 2018). The changes in society, economy, and environment have resulted in low levels of physical activity and long periods of sitting (Owen et al., 2020), which can increase the risk of non-communicable diseases such as type 2 diabetes, cardiovascular disease, musculoskeletal disability, and certain types of cancer.

The main focus of the current thesis was to investigate the potential of intelligent tutoring systems (ITSs) in training psychomotor skills, and to explore how these systems can be developed and utilized in a society that is increasingly concerned with global public health (McCuaig & Quennerstedt, 2018). The work presented is at the crossroad of two main fields, Artificial Intelligence in Education (AIED) and sports training. Our approach introduced an ontology in the psychomotor field, called OntoStrength, used by the ITS for knowledge modeling.

Also, based on the user's training constraints – limited time, fatigue, and volatile motivation, we introduced a tutoring personalization approach based on RL contextual multi-armed bandits, which was shown previously to be efficient in educational scenarios. Our simulations have shown promising results for using this approach in psychomotor training, and an experiment with real users has been performed between January and June 2022 which has shown the potential of the proposed method. Further experimentation is necessary to fully explore the use of ITSs for training psychomotor skills. The next section describes the personal contributions based on the three research questions formulated initially. Afterwards, some directions for future research are presented.

13.2 Personal Contributions

The development of multiple systems, and the contributions in various research areas – including computer science, knowledge modeling, or data science, all converging into a unified approach at the crossroad of informatics, educational science, and sports science, are just the highlighting points of the current thesis. The goal of our research was to understand which are the main requirements and implications for digitizing the psychomotor training using intelligent tutoring systems. Our work provides an inter-disciplinary approach, covering:

- *informatics*, with emphasis on reinforcement learning as support for adaptive tutoring, and computer engineering with the aim of developing the interface and communication modeling;
- *educational sciences*, for the intelligent tutoring systems study-cases, to discuss transferability and implications on the psychomotor field;
- *sports science*, for grounding the knowledgebase and developing the modeling using the ontological approach.

The preliminary work of this research was a systematic literature review on intelligent tutoring systems for psychomotor training which was published at the Intelligent Tutoring Systems

Conference³¹ in early 2020 and the work was cited and used by other researchers worldwide in similar projects. The thesis proposed further three systems, *OntoStrength*, *RiARiT*, and *Selfit*, which were developed by the author, under the direct guidance of both thesis supervisors: Sébastien Travadel and Răzvan Rughiniș. In terms of knowledgebase, the strength training field knowledge has been developed together with three members of the MINES ParisTech research center – Vincent Guarnieri, holding a master’s in sport sciences and sports training coach, and Eric Rigaud, researcher, holding a Ph.D. in computer science and athletics training coach, and Didier Delaitre – researcher, doctor in forensic medicine.

The developed knowledge by the sports specialists was further modeled and integrated within the Protégé software by Laurențiu Neagu and Eric Rigaud. The ontology was further exported in .owl format, shared across our team, and visualized initially using WebVOWL - Web-based Visualization of Ontologies, then using GraphDB. The initial version of the ontology has been published and accepted as a full paper at the GIFT Symposium in 2020³². An updated version has been developed in the years after, also a REST integration layer has been developed in the .NET Core technology, using C# programming language, which allows *OntoStrength* to interact with other systems. HTTP GET, POST, PUT, and DELETE methods have been developed to query, add, update, or delete ontology data. The work on the REST interface has been conducted by Laurențiu Neagu, with help from Emanuel Radu – an engineer in Computer Science. The updated *OntoStrength* has been published in late 2021 in Interaction Design and Architecture Journal in a special issue on Learning Ecosystems in the time of Covid-19³³.

The second contribution, *RiARiT*, was a method proposed for personalizing the learning sequence in psychomotor training. This method implied using Contextual Multi-Armed bandits’ algorithms for providing sports exercise recommendations while training. This is the first level of adaptation we envisioned – called Novice Trainer – and is using templates of training sessions which are filled in by exercise recommendations. Future levels of adaptation include generating personalized sessions on the fly – Intermediate Trainer -and creating personalized training plans, at microcycle, mesocycle, and macrocycle levels – Advance Trainer, which were not in the scope of the current thesis. *RiARiT* has been initially simulated with populations of virtual trainees and shown the potential of the approach. The simulations have been performed by Laurențiu Neagu with the close help of Mihai Dascălu – a full professor, with notable experience in automated systems and reinforcement learning.

³¹ <https://link.springer.com/book/10.1007/978-3-030-49663-0>

³² <https://gifttutoring.org/projects/gift/wiki/Overview>

³³ <http://ixdea.uniroma2.it>

This method has been published at the Intelligent Tutoring Systems Conference in 2021³⁴. *RiARiT* was further integrated into the tutoring component of a system for psychomotor development and tested with real users.

The third contribution, *Selfit*, was built on top of the previous two. *Selfit* is an intelligent tutoring system prototype that utilizes the knowledge model *OntoStrength*, and the personalization module *RiARiT* to exhibit the potential of such a system in psychomotor development. The User Interface module was developed entirely by the author of the thesis and, using ASP.NET MVC framework, and C# programming language. The development of the User Interface module started in early 2019 and many design and experience iterations have been conducted in the years after. Several features have been plugged in and out, and most of the effort was put into developing assessment modules. We initially implemented a Functional Movement Screening (FMS) test in *Selfit*, which implied users record themselves while performing the fundamental movements and the videos were sent to the cloud. An admin platform was developed in parallel, and sports coaches were able to label, from 0 to 3, the correctness of the executed movement.

Our team of sports scientists developed further a custom calibration protocol, more specific than the FMS test, which was implemented in the next version of the *Selfit* system. Also, in this version, users were not required to record themselves while performing the calibration, they could just manually input the perceived effort. Another assessment module implemented was the VO2Max Challenge, which is a test used to measure the VO2 capacity, a high important parameter in athletics training. This can be measured by constantly tracking trainee speed, and we performed the task using the GPS sensor, queried at small intervals on mobile devices, to compute the speed. We evaluated this module with real trainees, on a pitch, in an open field, and we were able to measure their VO2 capacity parameter. The *Selfit* prototype focus was on strength training, and so this assessment module was not integrated into the final version of the system. The data collected from wearables, from the most known manufacturers – Fitbit and Garmin, has been merged into the *Selfit* system. The users can synchronize their data on a dedicated page in the mobile application.

Also, a computer vision module has been implemented, separately from the *Selfit* system, to assess the risk of injuries for female athletes. This experiment used OpenCV³⁵ to track the movement angles and assess the risk of injuries. This implementation was not integrated within *Selfit*, as OpenCV cannot be added to the mobile application directly, and it does not work in real-time. It

³⁴ <https://link.springer.com/book/10.1007/978-3-030-80421-3>

³⁵ <https://opencv.org>

requires recorded, offline videos to make an analysis. For the scope of the experiment, we explored the potential of computer vision on psychomotor assessment, and we published our work as a full paper at the Smart Learning Ecosystems and Regional Development (SLERD) conference in 2021³⁶. This thesis was completed in collaboration with Vincent Guarnieri and Dănuț Matei, who have engineering degrees in computer science and a master's degree in computer network security.

The final refined version of *Selfit*, which was tested with real users between January-May 2022, with several features enriching the overall user experience (auto-mode, a chart with body area progress, home and gym specific training configuration, favorite exercises, roll/change an exercise, and others) was published for the third year in a row at the Intelligent Tutoring Systems Conference in 2022³⁷. To conclude, we consider that the initial goal to assess the requirements and implications of building an intelligent tutoring system for psychomotor development has been addressed through the multitude of learning tasks implemented and such a system has great potential in the field.

13.3 Directions for Future Research

One direction that requires further improvement is the accuracy of user assessment. The method implemented in the current thesis for assessing user progress relies on user manual input of the perceived effort. The number of repetitions in reserve is directly reported by each user. A method that has the potential to be more accurate for assessing the user effort per exercise is through the correlation of the *Selfit* current screen shown and the reported value of the heart rate through a wearable. Integration with main wearable manufacturers would be required and synchronization within the training session.

The variability of the heart rate is a good indicator of the perceived effort, both acute and chronic (short-term and long-term effects). This method can be used in conjunction with user input. We initiated the development for this module which currently permits the integration of Garmin and Fitbit data but does not make the correlation with the actual exercises in the session. Also, this module should be optional, and act only as an improvement in assessment accuracy for users who have such wearables and agree on sharing this sensitive data with us.

Another method for user assessment and also an improvement of the overall user experience is integrating a computer vision module. This module aims to measure the number of correct repetitions executed and automatically start the rest time chronometer and move to the next exercise

³⁶ <http://slerd2019.uniroma2.it>

³⁷ <https://link.springer.com/book/10.1007/978-3-031-09680-8>

when the required number of executions is completed. The computer vision feature is useful to compute the number of repetitions when the trainee performs less than the number of repetitions required. If the user can perform more than the number of repetitions shown in the interface, a wearable or user's manual input is required to assess the perceived effort.

Another area to focus on further is extending tutoring personalization from the Novice trainer to the Intermediate and Advanced trainer. The current version of the adaptive tutor can personalize the learning sequence at the exercise level, using pre-defined session templates. The Intermediate trainer should be able to discover optimal sessions (movements to train and order of exercises to train), generated by different sets of rules, but not pre-defined, and it should be used in conjunction with the Novice trainer. The Advanced trainer should be able to generate micro-cycles, mesocycles, and macrocycles for each user, and not use pre-defined templates. The Advanced trainer is based on user long-term feedback. The more advanced personalization modules will enable a fully customizable training experience for the users.

One improvement of the system proposed is the integration of the food dimension. Training programs should consider what people eat and drink. Recommendations on meal plans and nutrition, based on trainee profile, will improve training overall efficacy. The food dimension will require updates on knowledge modeling and the user interface module.

Selfit is currently used for strength training. Other training directions, such as flexibility or endurance, are also of high interest, especially for people with medical issues which prevent them from following a program for strength training. The development of the new training directions is already our focus

14 List of Publications

ISI Proceedings

1. Neagu, L.-M., Cotet, T.-M., Dascalu, M., Trausan-Matu, S., Badescu, L., & Simion, E. (2019a). Semantic Author Recommendations based on their Biography from the General Romanian Dictionary of Literature. In 7th Int. Workshop on Semantic and Collaborative Technologies for the Web, in conjunction with the 15th Int. Conf. on eLearning and Software for Education (eLSE 2019) (pp. 165–172). Bucharest, Romania: “CAROL I” National Defence University Publishing House. (WOS: 000473322400022) *
2. Neagu, L.-M., Rigaud, E., Travadel, S., Dascalu, M., Rughinis, R.-V. (2020). Intelligent Tutoring Systems for Psychomotor Training – A Systematic Literature Review. In: 16h Int. Conf. on Intelligent Tutoring Systems (ITS 2020) (pp. 335-341). Springer, Online. (WOS: 000720068400040)
3. Neagu, L.-M., Toma, I., Dascalu, M., Trausan-Matu, S., Hanganu, L., & Simion, E. (2020). A Quantitative Analysis of Romanian Writers’ Demography Based on the General Dictionary of Romanian Literature. In 5th Int. Conf. on Smart Learning Ecosystems and Regional Development (SLERD 2020) (pp. 253–261). Bucharest, Romania (Online): Springer. (WOS: 000783452900022) *
4. Neagu, L.-M., Rigaud, E., Guarnieri, V., Travadel, S., Dascalu, M. (2021). Selfit – An Intelligent Tutoring System for Psychomotor Development. In: 17th Int. Conf. on Intelligent Tutoring Systems (ITS 2021). Springer, Athens, Greece (Online). (WOS: 000718916000032)

BDI Proceedings

1. Neagu, L.-M., Cotet, T.-M., Dascalu, M., Trausan-Matu, S., Chisu, L., & Simion, E. (2019). Semantic Recommendations and Topic Modeling based on the Chronology of Romanian Literary Life. In 12th Int. Workshop on Social and Personal Computing for Web-Supported Learning Communities (SPeL 2019) held in conjunction with the 18th Int. Conf. on Web-based Learning (ICWL 2019) (pp. 164–174). Magdeburg, Germany: Springer *
2. Neagu, L.-M., Dascalu, M., Trausan-Matu, S., Chisu, L., & Simion, E. (2020). Automated Modeling of Romanian Literary Trends in History using Topics over Time and Co-Occurrences. In 8th Int. Workshop on Semantic and Collaborative Technologies for the Web, in conjunction with the 16th Int. Conf. on eLearning and Software for Education

(eLSE 2020) (pp. 151–158). Online: “CAROL I” National Defence University Publishing House. *

3. Neagu, L.-M., Guarnieri, V., Rigaud, E., Travadel, S., Dascalu, M., Rughinis, R.-V. (2020). An Ontology for Motor Skill Acquisition Designed for GIFT. Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8). Online.
4. Toma, I., Neagu, L.-M., Dascalu, M., Trausan-Matu, S., Hanganu, L., Simion, E.. (2020). Emerging Patterns in Romanian Literature and Interactive Visualizations based on the General Dictionary of Romanian Literature. In International Conference on Human-Computer Interaction (RoCHI2020): 91-103. Online: MatrixRom. *
5. Neagu, L.-M., Rigaud, E., Guarnieri, V., Matei, G.-D., Travadel, S., Dascalu, M.: Selfit – Accounting for Gender Differences in Personalized Motor Skills Learning. In: 6th Int. Conf. on Smart Learning Ecosystems and Regional Development (SLERD 2021). Springer, Bucharest, Romania (Online) (2021)
6. Neagu, L.-M., Rigaud, E., Guarnieri, V., Dascalu, M., Travadel, S.: Selfit v2 – Challenges Encountered in Building a Psychomotor Intelligent Tutoring System. In: 18th Int. Conf. on Intelligent Tutoring Systems (ITS 2022). Springer, Bucharest, Romania (Online)
7. Scheibenzuber, C., Neagu, L.-M., Ruseti, S., Artmann, B., Bartsch, C., Kubik, M., Dascalu, M., Trausan-Matu, S., Nistor, N.: Fake News Framing, Emotion, Argumentation, and Dialogic Social Knowledge Building in Online Discussions: An Exploration Including Natural Language Processing Data. In: 15th International Conference on Computer-Supported Collaborative Learning (CSCL) 2022. Hiroshima, Japan (Online)

Journals

ISI Journals

1. V. Pojoga, Neagu, L.-M., M. Dascalu (2020). The Character Network in Liviu Rebreanu’s Ion: A Quantitative Analysis of Dialogue. *Metacritic Journal for Comparative Studies and Theory*, 6, 2, 23–47. *
2. Neagu, L.-M., Rigaud, E., Guarnieri, V., Radu, E.I., Travadel, S., Dascalu, M., Rughinis, R. (2022). *OntoStrength: An Ontology for Psychomotor Strength Development*. *IxD&A (Interaction Design and Architecture (s)) Journal*, 52, 101-118.

Patents

1. Ruseti, S., Neagu, L.-M., Toma, I., & Dascalu, M. (2021). Metodă de Învățare Automată de Reprezentări Vectoriale în Grafuri de Cunoștințe pornind de la Modele de Limbă. Romania. Cererea A/100184 / 19.04.2021. OSIM. *

The publications marked with * are studies conducted within several research projects at Politehnica University of Bucharest, which are not directly linked with the current thesis

15 References

- Alessi, S. M., & Trollip, S. R. (2000). *Multimedia for learning: Methods and development* (3rd ed.). Boston, MA: Allyn & Bacon.
- Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2), 105-154.
- Ali Ahmed, G. H., & Kovacs, L. (2020). Ontology Domain Model for E-Tutoring System. *Software Engineering & Intelligent Systems*, 5(1).
- Almiyad, M. A., Oakden-Rayner, L., Weerasinghe, A., & Billinghamurst, M. (2017). *Intelligent augmented reality tutoring for physical tasks with medical professionals*. Paper presented at the Int. Conf. on Artificial Intelligence in Education, Wuhan, China.
- Andersen, P.-A., Kråkevik, C., Goodwin, M., & Yazidi, A. (2016). Adaptive task assignment in online learning environments. *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, 1-10.
- Aroyo, L., & Dicheva, D. (2004). The New Challenges for E-learning: The Educational Semantic Web. *Educational Technology & Society*, 7(4), 59-69.
- Arroyo, I., Beal, C. R., Murray, T., Waller, R., & Woolf, B. P. (2004). Wayang Outpost: Intelligent Tutoring for High Stakes Achievement Tests. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*. J. C. Lester, R. M. Vicari, and F. Paraguacu, Eds, Springer, Berlin, 142–169.
- Ausin, M. S., Azizoltani, H., Barnes, T., & Chi, M. (2019). Leveraging deep reinforcement learning for pedagogical policy induction in an intelligent tutoring system. *Proceedings of the Twelfth International Conference on educational data mining*, 168-177.
- Baron-Cohen, S., Knickmeyer, R. C., & Belmonte, M. K. (2005). Sex differences in the brain: implications for explaining autism. *Science*, 310(5749), 819-823.
- Barr, A., Beard, M., & Atkinson, R. C. (1976). The computer as a tutorial laboratory: The Stanford BIP Project. *International Journal of ManMachine Studies*, 8, 567–596. doi:10.1016/S0020-7373(76)80021-1
- Baumbach, S., Bhatt, A., Ahmed, S., & Dengel, A. (2018). Towards a Digital Personal Trainer for Health Clubs—Sport Exercise Recognition Using Personalized Models and Deep Learning. *Proceedings of the 10th International Conference on Agents and Artificial Intelligence*, 2, 438–445.
- Becker, H. J. (2000). Findings from the Teaching, Learning, and Computing Survey: Is Larry Cuban Right? *Education Policy Analysis Archives*, 8(51).
- Bilodeau, E., & Bilodeau, I. (1961). Motor-skills learning. *Annual Reviews of Psychology*, 12, 243–280.

- Biørn-Hansen, A., Majchrzak, T. A., & Grønli, T.-M. (2017). Progressive web apps: The possible web-native unifier for mobile development. *Proceedings of the 13th International Conference on Web Information Systems and Technologies, 1*, 344–351.
- Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York, NY, USA: David McKay Co Inc.
- Bompa, T., & Buzzichelli, C. (2017). Periodization: Theory and Methodology of Training. *Human Kinetics Publishers, Sixth édition*.
- Brawner, K., Hoffman, M., & Nye, B. (2019). *Architecture and Ontology in the Generalized Intelligent Framework for Tutoring: 2018 Update*. Paper presented at the 7th Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium.
- Brown, A. L., Collins, A., & Duguid, S. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32– 42.
- Brown, J. S., Burton, R. B., & deKleer, J. (1982). Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II, and III . Intelligent Tutoring Systems. *Sleeman and J.S.Brown (Eds) New York , Academic Press, 227 – 282*.
- Bruner, J. (1990). *Acts of meaning*. Cambridge, MA: Harvard University Press.
- Bull, S., & Mabbott, A. (2006). 20000 Inspections of a Domain-Independent Open Learner Model with Individual and Comparison Views. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems, Berlin Heidelberg, Springer-Verlag, 422–432*.
- 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), 77–97.
- Cao, Z. H., G., Simon, T., Wei, S.-E., & Sheikh, Y. (2018). OpenPose: realtime multi-person 3D pose estimation using Part Affinity Fields. *arXiv preprint arXiv:1812.08008*.
- Carbonell, J. R. (1970). AI in CAI: An Artificial-Intelligence Approach to ComputerAssisted Instruction. *IEEE Transactions on Man-Machine Systems, MMS-11*(4), 190-202.
- Chen, L., Chen, P., & Lin, Z. (2020). Artificial Intelligence in Education: A Review. *IEEE Access, 8*, 75264-75278.
- Chi, M. T. H., Feltovich, P., & Glaser, R. (1981). Categorization and Representation of Physics Problems by Experts and Novices. *Cognitive Science, 5*(2), 121– 152.
- Chidi-Ogbolu, N., & Baar, K. (2019). Effect of Estrogen on Musculoskeletal Performance and Injury Risk. *Front. Physiol., 9*. doi:10.3389/fphys.2018.01834

- Chughtai, R., Zhang, S., & Craig, S. D. (2015). Usability evaluation of intelligent tutoring system: ITS from a usability perspective. *Proceedings of the Human Factors and Ergonomics Society. Annual Meeting (HFES 2015)*, 367–371.
- Clement, B., Roy, D., Oudeyer, P.-Y., & Lopes, M. (2015). Multi-Armed Bandits for Intelligent Tutoring Systems. *Journal of Educational Data Mining (JEDM)*, 7.
- D Brown, B. B., B Goldberg. (2017). Authoring adaptive tutors for simulations in psychomotor skills domains. *In Proceedings of MODSIM World Wide Web*, 1-10.
- D. Hooshyar, R. B., Ahmad, M., Yousefi, F. D., & Yusop, S.-J. H. (2015). A flowchart-based intelligent tutoring system for improving problem-solving skills of novice programmers. *Journal of Computer Assisted Learning*, 31, 345-361.
- D’Mello, S., Craig, S., Gholson, B., Franklin, S., Picard, R., & Graesser, A. (2005). Integrating affect sensors in an intelligent tutoring system. *Proc. Computer in the Affective Loop Workshop at 2005 Int’l Conf. Intelligent User Interfaces*, 7-13.
- Das, D., Busetty, S. M., Bharti, V., & Hegde, P. K. (2017). Strength Training: A Fitness Application for Indoor Based Exercise Recognition and Comfort Analysis. *Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1126–1129.
- Dascalu, M., Westera, W., Ruseti, S., Trausan-Matu, S., & Kurvers, H. (2017). *ReaderBench Learns Dutch: Building a Comprehensive Automated Essay Scoring System for Dutch*. Paper presented at the 18th Int. Conf. on Artificial Intelligence in Education (AIED 2017), Wuhan, China.
- Dave, R. H. (1970). Psychomotor Domain. *Developing and Writing Behavioral Objectives*, 20–21.
- Dayan, P. (2009). Unsupervised learning. *The MIT Encyclopedia of the Cognitive Sciences*. Cambridge, MA, USA: MIT Press, 1–7.
- Dermeval, D., Albuquerque, J., Bittencourt, I. I., Isotani, S., Silva, A. P., & Vassileva, J. (2019). GaTO: An Ontological Model to Apply Gamification in Intelligent Tutoring Systems. *Frontiers Artificial Intelligence*, 2.
- Dermeval, D., Paiva, R., Bittencourt, I. I., Vassileva, J., & Borges, D. (2018). Authoring Tools for Designing Intelligent Tutoring Systems: a Systematic Review of the Literature. *International Journal of Artificial Intelligence in Education*, 28, 336–384.
- Devedzic, V., Jerinic, L., & Radovic, D. (2000). The GET-BITS Model of Intelligent Tutoring Systems. *Journal of Interactive Learning Research*, 11(3), 411-434.
- Diaz-Rodriguez, N., Wikstrom, R., Lilius, J., Cuellar, M. P., & Flores, M. D. C. (2013). *Understanding Movement and Interaction: An Ontology for Kinect-Based 3D Depth Sensors*. Paper presented at the Ubiquitous Computing and Ambient Intelligence. Context Awareness and Context-Driven Interaction (UCAmI 2013), Carrillo, Costa Rica.

- Dieste, O., Grimán, A., & Juristo, N. (2009). Developing search strategies for detecting relevant experiments. *Empirical Software Engineering*, 14(5), 513-539.
- Dlamini, M., & Leung, W. S. (2018). Evaluating Machine Learning Techniques for Improved Adaptive Pedagogy. *IST-Africa Week Conference, IST-Africa, IEEE*, 1.
- Dolenc, K., & Aberšek, B. (2015). TECH8 intelligent and adaptive e-learning system: Integration into technology and science classrooms in lower secondary schools. *Computers and Education*, 82, 354–365.
- Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching*. New York, NY, USA: Springer Heidelberg.
- Fensel, D. (2001). Ontologies. *Springer, Heidelberg*, 11-18.
- Fenza, G., & Orciuoli, F. (2016). Building pedagogical models by formal concepts analysis. *Proc 13th Int. Conf. Intelligent Tutoring Systems (ITS)*, 9684, 144-153.
- Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. *Proceedings AAAI Spring Symp. Series, AAAI Press, Menlo Park, California*, 33-40.
- Fister, I., Rauter, S., Yang, X. S., Ljubić, K., & Fister, I. (2015). Planning the sports training sessions with the bat algorithm. *149*, 993–1002.
- François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *CoRR*, 11(4), 219-354.
- Gay, G. (1986). Interaction of learner control and prior understanding in computer-assisted video instruction. *Journal of Educational Psychology*, 78, 225 – 227.
- Goldberg, B., Amburn, C., Ragusa, C., & Chen, D.-W. (2018). Modeling expert behavior in support of an adaptive psychomotor training environment: A marksmanship use case. *International Journal of Artificial Intelligence in Education*, 28(2), 194–224.
- Goodway, J. D., Ozmun, J. C., & Gallahue, D. L. (2019). *Understanding Motor Development: Infants, Children, Adolescents, Adults*. Burlington, MA, USA: Jones & Bartlett Learning.
- Graf, S. V., S.R., Kinshuk, & Leo, T. (2006). Representative Characteristics of Felder-Silverman Learning Styles: An Empirical Model. *Proceedings of the International Conference on Cognition and Exploratory Learning in Digital Age*, 235–242.
- Grawemeyer, B., Mavrikis, M., Holmes, W., Sergio, G. S., Wiedmann, M., & Rummel, N. (2016). Affecting Off-task Behaviour: How affect-aware feedback can improve student learning. *6th International Learning Analytics & Knowledge Conference*, 104-113.
- Grivokostopoulou, F., Perikos, I., & Hatzilygeroudis, I. (2017). An Educational System for Learning Search Algorithms and Automatically Assessing Student Performance. *International Journal of Artificial Intelligence in Education*, 27, 207–240.

- Guthold, R., Stevens, G. A., Riley, L. M., & Bull, F. C. (2018). Worldwide trends in insufficient physical activity from 2001 to 2016: a pooled analysis of 358 population-based surveys with 1.9 million participants. *Lancet Glob Health*, 6(10).
- Hackett, D. A., Johnson, N. A., Halaki, M., & Chow, C.-M. (2012). A novel scale to assess resistance-exercise effort. *Journal of Sports Sciences*, 30:13, 1405-1413. doi:10.1080/02640414.2012.710757
- Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). Fundamentals of item response theory. *Sage*, 2.
- Hassenzahl, M., Burmester, M., & Koller, F. . (2003). AttrakDiff : Ein Fragebogen zur Mes- sung wahrgenommener hedonischer und pragmatischer Qualität. J. Ziegler & G. Szwillus (Eds.) *Mensch & Computer 2003. Interaktion in Bewegung*, 187–196.
- Hodaie, Z., Haladjian, J., & Bruegge, B. (2018). *TUMA: Towards an Intelligent Tutoring System for Manual-Procedural Activities*. Paper presented at the International Conference on Intelligent Tutoring Systems, Montreal, Canada.
- Hoque, M. E. (2016). Three Domains of Learning: Cognitive, Affective and Psychomotor. *The Journal of EFL Education and Research (JEFLER)*, 2.
- ISO9241-210. (2019). Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems.
- Issurin, V. B. (2015). Building the Modern Athlete: Scientific Advancements and Training Innovations. *Muskegon Heights: Ultimate Training Concepts*.
- Joyce, D., & Lewindon, D. (2016). Sports Injury Prevention and Rehabilitation. Integrating medicine and science for performance solutions. *Taylor and Francis*.
- Katz, S., Lesgold, A., Hughes, E., Peters, D., Eggan, G., Gordin, M., & Greenberg, L. (1998). Sherlock 2: An intelligent tutoring system built on the LRDC framework. C. P. Bloom & R. B. Loftin (Eds.). *Facilitating the development and use of interactive learning environments*, 227–258.
- Kim, B., Suh, H., Heo, J., & Choi, Y. (2020). Ai-driven interface design for intelligent tutoring system improves student engagement. *Woodstock '18: ACM Symposium on Neural Gaze Detection*, 1-15.
- King, C., Kelvin, V., Warnars, H., Nordin, N., & Utomo, W. (2021). Intelligent Tutoring System: Learning Math for 6th-Grade Primary School Students. *Computational Intelligence in Education, 2021*.
- Kirchengast, S. (2010). Gender Differences in Body Composition from Childhood to Old Age: An Evolutionary Point of View. *Journal of Life Sciences*, 2(1), 1-10. doi:10.1080/09751270.2010.11885146
- Krathwohl, D. R., & Anderson, L. W. (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: New York : Longman.

- Lajoie, S., & Lesgold, A. (1992). Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship . *Intelligent instruction by computer: Theory and practice*. J.L. Farr and J. Psozka (Eds) , Washington D.C. , Taylor and Francis, 15– 36.
- LaViola, J., Williamson, B., Brooks, C., Veazanchin, S., Garrity, P., & Sottolare, R. (2015). Using Augmented Reality to Tutor Military Tasks in the Wild. *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, 1–10.
- Lee, D.-H., & Kim, J.-H. (2010). *A framework for an interactive robot-based tutoring system and its application to ball-passing training*. Paper presented at the 2010 IEEE Int. Conf. on Robotics and Biomimetics, Tianjin, China.
- Li, R. (2014). Why women see differently from the way men see? A review of sex differences in cognition and sports. *Journal of sport and health science*, 3(3), 155-162.
- Lieberman, J., & Breazeal, C. (2007). TIKL: Development of a wearable vibrotactile feedback suit for improved human motor learning. *IEEE Transactions on Robotics*, 23(5), 919–926.
- Liutsko, L., Muiños, R., Tous Ral, J. M., & Contreras, M. J. (2020). Fine Motor Precision Tasks: Sex Differences in Performance with and without Visual Guidance across Different Age Groups. *Behavioral Sciences*, 10(1), 36.
- Lu, T. P., D., & Pal, M. (2010). Contextual Multi-Armed Bandits. Proceedings of the 13th international conference on Artificial Intelligence and Statistics, 9, 485–492.
- Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology*, 106(4), 901.
- MacLellan, C. J., & Koedinger, K. R. (2020). Domain-general tutor authoring with apprentice learner models. *International Journal of Artificial Intelligence in Education*, 1–42.
- Mark, M. A., & Greer, J. E. (1993). Evaluation methodologies for intelligent tutoring systems. *Journal of Artificial Intelligence In Education*, 4, 129–153.
- Marouf, A., Abu Yousef, M., Mukhaimer, M., & Abu-Naser, S. (2018). An Intelligent Tutoring System for Learning Introduction to Computer Science. *International Journal of Academic Multidisciplinary Research (IJAMR)*, 2(2), 1-8.
- Martha, A. S. D., & Santoso, H. B. (2019). The design and impact of the pedagogical agent: A systematic literature review. *Journal of Educators Online*, 16.
- McCuaig, L., & Quennerstedt, M. (2018). Health by stealth—exploring the sociocultural dimensions of salutogenesis for sport, health and physical education research. *Sport, education and society*.
- McGuinness, D. L., & Van Harmelen, F. (2004). OWL Web Ontology Language overview. *W3C recommendation*, 10(10), 2004.

- McNamara, D. S., Crossley, S. A., & Roscoe, R. (2013). Natural language processing in an intelligent writing strategy tutoring system. *Behavior research methods*, 45(2), 499-515.
- Me, E., & Unold, O. (2011). Machine learning approach to model sport training. *Comput. Hum. Behav.*, 27(5), 1499-1506.
- Melesko, J., & Novickij, V. (2019). Computer adaptive testing using upper-confidence bound algorithm for formative assessment. *Appl. Sci.*(20), 4303.
- Mergel, B. (1998). Instructional design and learning theory. *Educational communications and technology*, 1-24.
- Mohamed, H., & Lamia, M. (2018). Implementing flipped classroom that used an intelligent tutoring system into learning process. *Computers & Education*, 124, 62-76.
- Mohammed, P., & Mohan, P. (2015). Dynamic cultural contextualisation of educational content in intelligent learning environments using ICON. *International Journal of Artificial Intelligence in Education*, 25, 249–270.
- Mota, S., & Picard, R. W. (2003). Automated Posture Analysis for Detecting Learner's Interest Level. *Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction, CVPR HCI*.
- Mousavinasab, E., Zarifsanaiy, N., Niakan Kalhori, S. R., Rakhshan, M., Keikha, L., & Ghazi Saeed, M. (2018). Intelligent tutoring systems: A systematic review of characteristics, applications, and evaluation methods. *Interact. Learn. Environ.*, 1-22.
- Mui, J., Lin, F., & Dewan, M. (2021). Multi-armed Bandit Algorithms for Adaptive Learning: A Survey. *International Conference on Artificial Intelligence in Education*, 273–278.
- Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 98-129.
- Neagu, L.-M., Guarnieri, V., Rigaud, E., Travadel, S., Dascalu, M., & Rughinis, R.-V. (2020). An Ontology for Motor Skill Acquisition Designed for GIFT. *Proceedings of the 8th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym8)*, 37-47.
- Neagu, L.-M., Rigaud, E., Travadel, S., Dascalu, M., & Rughinis, R.-V. (2020). *Intelligent Tutoring Systems for Psychomotor Training – A Systematic Literature Review*. Paper presented at the 16h Int. Conf. on Intelligent Tutoring Systems (ITS 2020), Online.
- Nkambou, R., Bourdeau, J., Mizoguchi, R. (2010). Introduction: What Are Intelligent Tutoring Systems, and Why This Book. *Advances in Intelligent Tutoring Systems*, 1-12.
- Noy, N. F., Crubezy, M., Ferguson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). *Protege-2000: an open-source ontology-development and knowledge-acquisition environment*. Paper presented at the AMIA Annual Symposium, Washington, DC, USA.

- Noy, N. F., & McGuinness, D. L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Retrieved from Stanford, CA, USA:
- Orey, M. A., Nelson, W.A. (1993). Development Principles for Intelligent Tutoring Systems: Integrating Cognitive Theory into the Development of Computer-Based Instruction. *Association for Educational Communications and Technology*, 41.
- Ota, M., Tateuchi, H., Hashiguchi, T., Kato, T., Ogino, Y., Yamagata, M., & Ischihashi, N. (2020). Verification of reliability and validity of motion analysis system during bilateral squat using human pose tracking algorithm. *Gait & Posture* 80, 62-67.
- Owen, N., Healy, G. N., Dempsey, P. C., Salmon, J., Timperio, A., Clark, B. K., . . . Lambert, G. (2020). Sedentary behavior and public health: integrating the evidence and identifying potential solutions. *Annu. Rev. Public Health*, 41 265-287.
- P. Andreatta, P. D. (2019). Supporting the Development of Psychomotor Skills. *Advancing Surgical Education*, 183-196.
- Paladines, J., & Ramirez, J. (2020). A systematic literature review of intelligent tutoring systems with dialogue in natural language. *IEEE Access*, 8, 164246–164267.
- Panagiotopoulos, I., Kalou, A., Pierrakeas, C., & Kameas, A. (2012). An Ontology-Based Model for Student Representation in Intelligent Tutoring Systems for Distance Learning. *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 296-305.
- Paviotti, G., Rossi, P.G., Zarka, D. (2012). Intelligent tutoring systems: an overview. *Pensa Multimedia*.
- Pitchers, G., & K., E.-S. (2019). Considerations for coaches training female athletes. *PROFESSIONAL STRENGTH & CONDITIONING, Training Female Athletes*, 19-30.
- Polson, M. C., & Richardson, J. J. (1988). Foundations of Intelligent Tutoring Systems. *Lawrence Erlbaum Associates, Hillsdale*, 79-108.
- PRB de Campos, E. N., UF Moreno. (2018). Proposal of a new taxonomy of the psychomotor domain for to the engineering laboratory. *3rd International Conference of the Portuguese Society for Engineering Education*.
- Prud'hommeaux, E., & Seaborne, A. (2008). SPARQL query language for RDF. W3C Recommendation. In: World Wide Web Consortium.
- Rajšp, A., & Fister, I. (2020). A systematic literature review of intelligent data analysis methods for smart sport training. *Appl. Sci.*, 10, 3013.
- Ramkumar, S., & Poorna, B. (2017). Development of Ontology for Sports Domain. *International Journal for Research in Applied Science & Engineering Technology*, 5.

- Ropelato, S., Zund, F., Magnenat, S., Menozzi, M., & Sumner, R. (2018). Adaptive tutoring on a virtual reality driving simulator. *1st Workshop on Artificial Intelligence Meets Virtual and Augmented Worlds (AIVRAR) in Conjunction with SIGGRAPH Asia*.
- Rosé, C. P. (2000). A Framework for Robust Semantic Interpretation. Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics. San Francisco, CA, Morgan Kaufmann, 311–318.
- Rygula, I. (2005). Artificial neural networks as a tool of modeling of training loads. *Proceedings of the 2005 IEEE engineering in medicine and biology 27th annual conference*, 1-4.
- Samarakou, M., Prentakis, P., Mitsoudis, D., Karolidis, D., & Athinaios, S. (2017). Application of fuzzy logic for the assessment of engineering students. *IEEE Global engineering education Conference, EDUCON*.
- Schez-Sobrino, S., Gmez-Portes, C., Vallejo, D., Glez-Morcillo, C., & Redondo, M. Á. (2020). An Intelligent Tutoring System to Facilitate the Learning of Programming through the Usage of Dynamic Graphic Visualizations. *Appl. Sci.*, *10*. doi:doi:10.3390/app10041518
- Schulze, K. G., Shelby, R. N., Treacy, D. J., Wintersgill, M. C., Vanlehn, K., & Gertner, A. (2000). Andes: An intelligent tutor for classical physics. *The Journal of Electronic Publishing*, *6*(1).
- Self, J., & Akhras, F. (2002). Beyond intelligent tutoring systems: situations, interactions, processes and affordances. *Instructional Science*, *30*, 1-30.
- Selfridge, O. (1993). From the Gardens of Learning: A Vision for AI. *AI Magazine*, *14*, 36– 48.
- Semerci, Ç., & Batdi, V. (2015). A Meta-Analysis of Constructivist Learning Approach on Learners' Academic Achievements, Retention and Attitudes. *Journal of Education and Training Studies*, *3*(2), 171-180.
- Shapiro, S. (1992). Encyclopedia of Artificial Intelligence, 2nd Edition. *John Wiley & Sons*.
- Shephard, R. J. (2000). Exercise and training in women, Part I: Influence of gender on exercise and training responses. *Can J Appl Physiol*, *25*(1), 19-34. doi:10.1139/h00-002
- Shute, V., & Psotka, J. (1995). Intelligent tutoring systems: Past, Present, and Future. *Handbook of Research on Educational Communications and Technology*, : 570–600.
- Simon, T., Joo, H., Matthews, I., & Sheikh, Y. (2017). Hand Keypoint Detection in Single Images Using Multiview Bootstrapping. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1145-1153.
- Simpson, E. J. (1966). The classification of educational objectives, psychomotor domain. *Illinois Journal of Home Economics*, *10*, 110-144.

- Skinner, A., Diller, D., Kumar, R., Cannon-Bowers, J., Smith, R., Tanaka, A., . . . Perez, R. (2018). Development and application of a multi-modal task analysis to support intelligent tutoring of complex skills. *International journal of STEM education*, 5(1), 1-17.
- Somerson, J. S., Isby, I. J., Hagen, M. S., Kweon, C. Y., & Gee, A. O. (2019). The Menstrual Cycle May Affect Anterior Knee Laxity and the Rate of Anterior Cruciate Ligament Rupture: A Systematic Review and Meta-Analysis. *JBJS Rev.*, 7(9). doi:10.2106/JBJS.RVW.18.00198
- Sottolare, R., Graesser, A., Hu, X., & Holden, H. (2013). Design recommendations for Intelligent Tutoring Systems: learner modeling (vol. 1). *Orlando, FL: U.S. Army Research Laboratory*.
- Sottolare, R. A., Brawner, K. W., & Sinatra, A. M. (2017). An updated concept for a Generalized Intelligent Framework for Tutoring (GIFT). *GIFTtutoring.org*.
- Spielholz, P. (2006). Calibrating Borg scale ratings of hand force exertion. *Applied Ergonomics*, 37, 615-618.
- Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. *MIT Press*, 398.
- Sweller, P., & Chandler, S. (1994). Why some material is difficult to learn. *Cognition and Instruction*, 12(3), 185–233.
- Taub, M., Azevedo, R., Rajendran, R., Cloude, E. B., Biswas, G., & Price, M. J. (2019). How are students' emotions related to the accuracy of cognitive and metacognitive processes during learning with an intelligent tutoring system? *Learning and Instruction*.
- Urban-Lurain, M. (1996). Intelligent tutoring systems, an historic review in the context of the development of artificial intelligence and educational psychology. *Technical report, Department of Computer Science and Engineering, Michigan State University*.
- VanLehn, K. (1988). Student Modeling. *Foundations of Intelligent Tutoring Systems*, Hillsdale, New Jersey: Lawrence Erlbaum Associates Publishers, 55–78.
- VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46, 197–221. doi:10.1080/00461520.2011.611369
- Vygotsky, L. S. (1978). *Mind in society*. Cambridge, MA: Harvard University Press.
- Weragama, D., & Reye, J. (2014). Analysing student programs in the PHP intelligent tutoring system. *International Journal of Artificial Intelligence in Education*, 24, 162–188.
- Woolf, B. P. (2010). Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing e-Learning. *Morgan Kaufmann Publishers*.
- Zhou, B., Sundholm, M., Cheng, J., Cruz, H., & Lukowicz, P. (2016). Never skip leg day: A novel wearable approach to monitoring gym leg exercises. *Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 1-9.

Zouaq, A., & Nkambou, R. (2010). A survey of domain ontology engineering: Methods and tools. *Advances in Intelligent Tutoring Systems, Springer-Verlag, Berlin Heidelberg*, 103–119.

RÉSUMÉ

La thèse étudie la contribution des « Intelligent Tutoring Systems » (ITS) à l'évaluation et au développement des compétences psychomotrices. Après avoir démontré la valeur ajoutée à l'utilisation des ITS pour le développement des compétences psychomotrices, les défis technologiques associés ont été identifiés et des solutions proposées. La contribution de la thèse est la conception, le développement et la validation des systèmes OntoStrength, RiARit et Selfit. OntoStrength est une ontologie dédiée aux compétences psychomotrices développée avec le soutien de spécialistes du sport. RiARiT est un algorithme d'individualisation des séquences d'apprentissage pour le développement psychomoteur, et Selfit est un prototype d'ITS dédié au développement psychomoteur.

MOTS CLÉS

Système de tutorat intelligent, habilités psychomotrices, ontologie, apprentissage par renforcement

ABSTRACT

The thesis studies the potential contribution of Intelligent Tutoring Systems (ITSs) in training psychomotor skills. Technological challenges were defined and addressed after demonstrating the benefits of using ITS in psychomotor training. The thesis's main contribution is designing, developing, and validating the OntoStrength, RiARit, and Selfit systems. OntoStrength is an ontology dedicated to psychomotor competencies developed with the support of sports specialists. RiARiT is an algorithm for personalizing learning sequences in psychomotor training, and Selfit is the prototype of an Intelligent Tutoring System dedicated to psychomotor development. A validation campaign demonstrates the added value of these systems and publications contribute to multiple fields, including computer science, knowledge modeling, data science, educational science, and sports science.

KEYWORDS

Intelligent Tutoring system, Psychomotor abilities, Ontology, Reinforcement learning