

AL AKHAWAYN
UNIVERSITY

University Honors Program

Capstone Design

Najwa Laabid

Dr. Violetta Cavalli-Sforza

April 2019

INTELLIGENT TUTORING SYSTEM FOR EARLY ALGEBRA

Honors Capstone Report

I, Najwa Laabid, hereby affirm that I have applied ethics to the design process and in the selection of the final proposed design. I have also held the safety of the public to be paramount and have addressed this in the presented design wherever applicable.



Approved by the Supervisor(s)



ACKNOWLEDGEMENTS

I would like to thank Dr. Violetta Cavalli-Sforza for agreeing to supervise my capstone despite her being on a sabbatical. This work would not have been possible without her continuous guidance, extensive support, and multiple shared resources throughout the project. I would like to thank her also for her mentorship and career advice in the field of educational technology, both as a supervisor and an inspiring faculty member.

I would like to express my gratitude to my parents and family, who showed me nothing but encouragement throughout my journey in AUI. Many of the experiences I have had the chance to live in the past five years, which contributed directly or indirectly to the fruition of this project, would not have been possible without the material and emotional support of my immediate family. I would like to thank specifically my dad, Ezzaim Laabid, and his friend Mr. Mustapha Ourahay, for sharing their expertise in teaching algebra as pedagogues in the field.

I am grateful to Pr. Abdelkrim Daoudi, my 8th grader math teacher who kindly agreed to share his experience in teaching early algebra following the Moroccan curriculum. I would like to thank him for his efforts as a serious and competent math teacher 10 years ago and a valuable resource to this project today. I extend my gratitude as well to Mr. Ahmed Ibrahimi who participated to the same interview as Mr. Daoudi on a short notice and out of sheer interest in the project.

Finally, I would like to thank my friends who made AUI a second home, the faculty who mentored and taught me throughout my undergraduate journey, and the professionals I met in industry and academia who helped me discover my passion for educational technology through long discussions and direct or indirect mentorship.

Table of Contents

ACKNOWLEDGEMENTS	III
LIST OF FIGURES.....	V
LIST OF TABLES.....	VI
LIST OF ACRONYMS AND ABBREVIATIONS.....	VII
ABSTRACT	VIII
1 INTRODUCTION.....	1
2 PEDAGOGICAL RESEARCH	2
2.1 INTRODUCTION	2
2.2 LITERATURE REVIEW	3
2.3 RESULTS: ITS CURRICULUM	12
3 TECHNICAL RESEARCH.....	15
3.1 INTRODUCTION.....	15
3.2 LITERATURE REVIEW	15
3.3 RESULTS: TECHNICAL CHOICES FOR THE ITS	18
4 OVERVIEW OF CTAT	19
4.1 INTRODUCTION	19
4.2 COMPONENTS OF CTAT	19
4.3 RESULTS: TECHNICAL DESIGN OF THE SYSTEM	26
5 ITS: SYSTEM DESCRIPTION	27
5.1 INTRODUCTION.....	27
5.2 OVERVIEW OF THE SYSTEM COMPONENTS.....	27
5.3 DEPLOYMENT AND TESTING	33
6 STEEPLE ANALYSIS.....	36
CONCLUSION AND FUTURE WORK.....	39
REFERENCES.....	40
APPENDIX A: PROFESORS' INTERVIEW SUPPORT SHEET (ENGLISH VERSION).....	42
APPENDIX B: ITS STUDENT INTERFACES	47

LIST OF FIGURES

Figure 1: Algebra Mal-Rules suggested by Squibb and Payne [5].....	9
Figure 2: Excerpt from Moroccan Textbook Showcasing the Use of the Segments Method for Problem-Solving	11
Figure 3: MathTutor Problem-Solving Module Using the Segment Method.....	11
Figure 4: HTML Editor Login Page	20
Figure 5: HTML Editor in the Process of Creating a Student Interface	20
Figure 6: Folder Structure of a CTAT Package.....	21
Figure 7: CTAT HTML and Flash Software Set to Demonstrate Author Mode.....	22
Figure 8: Example of a Behavior Graph	23
Figure 9: Jess Console on CTAT for HTML and Flash.....	25
Figure 10: Working Memory Editor Window from CTAT for HTML and Flash	25
Figure 11: Module 2 Student Interface	28
Figure 12: Module 3 Student Interface (1).....	29
Figure 13: Module 3 Student Interface (2).....	29
Figure 14: Sample Graph from Module 2	30
Figure 15: Sample Rule from Module 2.....	31
Figure 16: Email Granting Permission to Use TutorShop	34
Figure 17: Welcome Page of TutorShop.....	34
Figure 18: First Problem of Module 2 Hosted on TutorShop	35
Figure 19: Interface for Module 1 Problem 1.....	47
Figure 20: Interface for Module 1 Problem 2.....	48
Figure 21: Interface for Module 1 Problem 3.....	48
Figure 22: Interface for Module 1 Problem 4.....	49

LIST OF TABLES

Table 1: ILO for Early Algebra Education (Literature Review).....	6
Table 2: Module 1 Sub-Skills, Strategies and Expected Difficulties	14
Table 3: Skills Captured by the Behavior Recorder per Module	33
Table A1: Algebra ILO for Interview Support Sheet	43
Table A2: Notes on Proposed Algebra Strategies	44
Table A3: Arithmetic Prerequisites for Teaching Algebra	45

LIST OF ACRONYMS AND ABBREVIATIONS

ITS Intelligent Tutoring System

ILO Intended Learning Outcome(s)

CTAT Cognitive Tutor Authoring Tool

CMU Carnegis-Mellon University

ABSTRACT

This capstone project is an intelligent tutoring system for teaching introductory algebra. The aim of the system is to assist middle-school students (between 6th and 7th grade) in the development of algebraic thinking, by offering an adaptive environment to practice solving first degree equations with one unknown and modeling real-world problems with equations of the same type. The project will provide an overview of the state-of-the-art pedagogy of early-algebra education, including recommended strategies for the teaching of the aforementioned skills, and common errors and misconceptions in the process of transitioning from arithmetic to algebra. These findings will be synthesized in the curriculum of the system, which will remain as analogous as possible to the Moroccan middle-school math curriculum of the first year of middle school (7th grade). Finally, an adaptive educational environment is implemented, using the Cognitive Tutor Authoring Tool (CTAT) offered by Carnegie-Mellon University (CMU), to provide students with an optimal learning experience in a field crucial to their subsequent mathematical education.

Keywords: intelligent tutoring system, early algebra, mathematics education, cognitive tutor

1 INTRODUCTION

There seems to be a consensus in the educational literature on the importance of introductory algebra in subsequent mathematical education. The transition from arithmetic, defined as operating on numbers, to algebra, which is operating on unknown entities and objects, is a crucial step in the development of high-level mathematical thinking, and one which seems to be problematic for a large number of children.

The same sources also seem to agree on the usability of one-to-one education. Personalized instructions and feedback are known to make a large positive impact on the learning efficiency of students [1]. Individual tutoring is not often affordable to families, particularly in underprivileged areas. Even with no financial concerns in sight, individual human instruction is the least scalable educational model of all – there simply are not enough teachers to afford dismantling the schooling system as we know it today. With the advancement of technology, particularly artificial intelligence, the possibility of making of machines personalized instructors became an alternative worthy of exploration and potential investment. Intelligent Tutoring Systems (ITSs) emerged therefore as a possible solution to optimize learning experience at an affordable cost. [1]

This capstone project explores the creation of an ITS operating particularly on early-algebra lessons. The aim of the project is to create an adaptive environment to teach students the resolution of first-degree equations with one unknown and the modeling of mathematical problems with equations of the same type. The final product would be an online-system fulfilling the aforementioned purpose. Future work on the project would include testing the final product with a real audience and making any adjustments deemed fit to improve its performance.

2 PEDAGOGICAL RESEARCH

2.1 INTRODUCTION

I knew I wanted to make an ITS for K-12 mathematical education about a year ago. Before moving further, it was necessary to narrow the scope of the subject to a specific lesson for the sake of time and to increase the quality of the project. Deciding on a single lesson from the wide range of topics covered within the chosen school years required a considerable amount of browsing through school curricula, to eventually come down to a matter of personal preference. In fact, while I was looking at one particular curriculum developed by Ed4.0¹ to assess numerical literacy among Moroccan children, problem solving exercises automatically caught my attention. This specific type of mathematical applications has the double advantage of testing a wide range of skills through a single problem and offering a chance to add relevance to mathematical concepts, which may remain confined in abstraction otherwise.

Following this line of thought, I realized that I was particularly interested in how equations are used as a modeling and problem-solving tool. I remembered from my own first exposure to algebra in 7th grade my fascination and struggle with first-degree equations and the related concept of ‘literal calculation’². Getting used to algebraic thinking and methodology takes time and effort, and may never happen for some pupils, to whom math becomes alien as soon as letters take over what was initially a numbers-only domain [2]. This reported difficulty in expanding children’s mathematical horizons is perhaps intensified by the centrality of basic algebra in mathematical education starting 7th grade and onward. From the possibility of abstracting mathematical concepts in proofs, to the power of modeling, explaining and interpreting unleashed by equations, basic algebra is perhaps *the* key element in any mathematical tool kit. Beside arithmetic, math lays ahead of algebra.

Once the importance and intrigue of early algebra became apparent to me, I moved on to define the pedagogical context of the ITS, in an attempt to answer the following general question: *how can an intelligent tutor best teach algebra?* The answer to this question would come in the form

¹ Ed4.0 is a rising international NGO operating in the field of educational technology. I was introduced to their work through an event in AUI in Fall 2017.

² The expression ‘literal calculation’ is my own translation of the phrase ‘calcul littéral’ in French or ‘الحرفي الحساب’ in Arabic.

of a personal curriculum meant to introduce and train students on the topic of first-degree equations with one unknown. This topic is typically the first lesson of most algebra curricula because of its simplicity and applicability to the type of problems students are familiar with through earlier math classes. Before answering the ‘how’ of teaching algebra however, it was important to define the ‘what’ of the lessons, i.e. the Intended Learning Outcomes (ILOs) of early algebra education. To devise a suitable and effective curriculum, it is also crucial to specify the pre-requisites of studying algebra and determine ways to ensure their existence before tackling the new topic. An awareness of the difficulties and errors commonly experienced by students when performing algebraic manipulations is also necessary to implement the real-time feedback and assessment features expected of an ITS. Finally, all this literature work should be projected onto the Moroccan curriculum to make the system better fit its intended audience.

This section attempts to answer the aforementioned questions through a literature review of the state-of-the-art research on algebra education, including a review of the Moroccan 7th grade curriculum and how it meets (and sometimes fails to meet) the findings of the research. The section concludes with a description of the curriculum devised for this ITS in light of the pedagogical study.

2.2 LITERATURE REVIEW

2.2.1 INTENDED LEARNING OUTCOMES OF EARLY ALGEBRA EDUCATION

Before discussing ILOs, it is important to define the term algebra, and more specifically ‘early algebra’, within the scope of this project. Algebra ‘pedagogically’ can be seen in contrast with its most direct ancestor in mathematical education: arithmetic. Arithmetic is the study of numbers and their manipulation. From primary to middle school, this field covers topics such as basic types of numbers (integers, decimals, fractions, negative numbers, etc.), the four elementary operations (addition, subtraction, division, and multiplication), and a few relatively advanced topics such as factorization and distributing an operation on another one [2]. Arithmetic can also be credited for introducing fundamental calculation and number properties, such as evenness/oddness, precedence of arithmetic operations, commutativity, associativity, etc. In short, everything within the ‘sphere’ of numbers is the domain of arithmetic in the context of this research.

Algebra begins when entities other than numbers are added onto arithmetic. The presence of letters in mathematical expressions (e.g. $n+2$, $3a+4b$, etc.) signals algebra territory, despite the expressions involving the same properties and/or concepts used in arithmetic. In fact, although algebra can be seen in this context as a natural extension of arithmetic, the transition does not happen as naturally as educators would hope [2]. Let us look at addition from an algebraic point of view for instance, using the following expression as an example: $a+a-b+3a$. Simplifying (or really computing) this sum requires grouping and merging similar terms and updating the coefficient next to the corresponding symbol. In other words, the addition happens at the level of coefficients, and requires some implicit knowledge on working with variables, such as the fact that a is a short version of $1a$, and that an addition between two variables a and b cannot be further simplified (same applies for a number and a variable, like $2a + 3$).

When looking closely at all other arithmetic operations in their algebraic form, it becomes apparent that expanding arithmetic to algebra requires introducing new knowledge perhaps more than it relies on arithmetic concepts. The novelty brought by algebra stands out even more in its second fold: equations. With the ability to operate on almost anything now, it is necessary to upgrade problem solving to include more powerful and methodic tools [2]. Beside the challenge of learning a new method, manipulating equations requires its own set of algebra specific concepts and rules. From here, we can define algebra concisely as operating on entities other than numbers to solve problems or equations.

One final clarification is owed to introduce the specific focus of this project: *early* algebra. The qualification is used mostly to emphasize the simplicity of the topics covered by this research and implemented by the subsequent system. In other words, we are not looking at the full domain of algebra, but rather at a specific skillset considered to be at the transition point from arithmetic to algebra. For the remainder of this report, '*early*' and '*basic*' would be the designated qualifiers for emphasizing the simplicity of the lessons of the ITS. Early algebra is not to be confused with pre-algebra, which denotes in literature informal methods used in mathematical education to introduce students to problem solving [3].

The ILOs of algebra follow naturally from the discussion above. In general, within this domain, students are expected to: manipulate algebraic expressions, model and solve word problems, and apply algebra to prove theorems and relations. Table 1 provides a summary of the subskills

of each ILO along with its description. It is worth mentioning that this particular project will focus on the first two ILOs since it is implementing the Moroccan 7th grade curriculum which does not introduce algebraic proofs. More on this topic can be found in the section dedicated to the analysis of the Moroccan curriculum.

2.2.2 KEY CONCEPTS AND STRATEGIES TO TEACH THEM

Meeting the ILOs mentioned earlier requires the assimilation of a set of key concepts crucial to the understanding and/or application of each of the aforementioned skills. This section presents one key concept per ILO. Since there are no hard boundaries between the ILOs, these concepts can be seen as central to algebra education in general, beyond the skills' framework defined in the previous section.

Concept 1: Literal Representation of Variables in Mathematical Expressions

Performing literal calculations requires a familiarity with the use of literal symbols to represent unknown and/or general entities (i.e., variables). Once the student is aware that those novel symbols are not in their essence much different from the numbers he or she is used to, they should be encouraged to project the number properties they are familiar with onto those new entities. This is relevant for both arithmetic manipulations and algebraic proofs. To introduce arithmetic operations on literals, it may help to remind students of the original notions associated with these operations. Many arithmetic operations were after all introduced using contextual data: an addition is a counting of objects before it is an abstract numerical operation, a multiplication is a short notation of repetition, a division translates the relationships of parts to a whole, etc. Leveraging the previous knowledge of students in this way, with the minimal addition of literal notations as a single novel concept, can help smoothen the transition from arithmetic to algebra.

Concrete Strategies:

Statements such as 'Let us use x to symbolize the number we are looking for' generally suffice for the initial exposure to this concept. It is advisable to make this introduction in problems students are familiar with to downplay the factor of novelty. In some special cases, making use of real-life objects helps illustrate arithmetic operations and notions on the difference between variables in a single expression [4]. Finally, practice on literal

calculation of gradual difficulty is paramount to the mastery of this skill, which is central to all remaining ILO.

Table 1: ILO for Early Algebra Education (Literature Review)

Intended Learning Outcome	Subskills
Algebraic Manipulations	<ul style="list-style-type: none"> - Ability to perform arithmetic operations on variables represented by letters: <ul style="list-style-type: none"> • Recognize different variables and the operations allowed between them (e.g., it is possible to multiply a variable entity by a number: $2 * (3a) = 6a$, but not to add a number to a variable entity: $2 + 3a \neq 2 + 3a$). • Simplify long algebraic expressions using basic and advanced operations (including distribution and factorization). - Manipulating equations: <ul style="list-style-type: none"> • Operating on both sides to get an equivalent equation. • Operating on various terms of the equations with the aim of solving for an unknown value.
Modeling and solving contextual problems	<ul style="list-style-type: none"> - Translating relevant information from a text to a mathematical equation. - Manipulating the equation to find the unknown value. - Interpreting the result found in the previous step within the context of the problem.
Simple algebraic proofs	<ul style="list-style-type: none"> - Making use of abstract mathematical notations to prove a numerical theorem or relation (e.g., to show that the sum of three consecutive integers is always divisible by three, a student should make use of the expression $n + n + 1 + n + 2 = 3n + 3 = 3k$). - Recognize and represent the conditions within which the proof is valid (n and k in the example above are integer values. The relation does not hold for real numbers).

Concept 2: Algebraic Equality

Arithmetic operations are not the only concepts seemingly shared between algebra and arithmetic, despite containing enough novelty to become a source of confusion when badly-introduced. In fact, research shows that algebraic equality, particularly its symmetry in the context of equation resolution, does not come intuitively to students [2]. The research reports

an interesting finding regarding the pre-symbolic algebra methods for equation resolution between the 13th and the 15th century. At that time, solving the two equations $x^2+c = 2bx$ and $2bx = x^2+c$ used completely different approaches, which suggest an unfamiliarity with the symmetry of the equal sign [2].

In current day mathematical education, students use equations to express a cause-consequence relationship throughout their study of arithmetic, be it in calculation or problem solving (e.g., $3+5$ (cause) = 8 (consequence/result)). Manipulating algebraic equations, however, requires looking at the two sides of the equality as equivalent expressions which should preserve their equivalence throughout the resolution process. Assimilating this idea is particularly relevant to handling equations where the unknown appears on both sides (equation of the form $ax+b = cx+d$), in which unwinding strategies³ do not suffice. It is therefore recommended to ‘break’ the causality of the equal sign inherited from arithmetic as part of algebraic initiation [2].

Concrete Strategies:

Students should already be familiar with the symmetry of equality in numbers. This knowledge needs to be leveraged to introduce the notion of equivalence in the context of equations. To this aim, variations can be used in writing equations derived from problems, such as purposefully reverting the right- and left-hand sides of the equation and asking students to solve the equation in its new form. Using analogies and external tools (like assimilating the equation to sides of a balance and equivalence to an equilibrium state) is also likely to ease the understanding of this concept [2].

Concept 3: Translation of Text into Mathematical Expressions

Even after mastering the manipulation of algebraic equations, applying them to problem resolution still poses the challenge of modeling data in literary form. Expressing the conditions and relationships present in a text using mathematics is half the work when solving problems. In order to help students develop this skill, it is important to introduce it in its most basic form following the strategies suggested below.

³ Unwinding strategies are an informal way to approach simple contextual problems and/or equation, by thinking of the simplification process in terms of operations to “undo”. For example, to find the value of x in $x+4=5$, we reverse the addition by performing a subtraction of the two known entities 5 and 4, to get -1 . This type of manipulations does not require an understanding of algebraic properties.

Concrete Strategies

Creating the final mathematical equation of a problem requires first identifying relevant relations and conditions. Highlighting these for students might shed light onto how parts of the equation are extracted from the text to form the bigger picture. Another recommended method for problem solving using algebraic tools is to enforce a systematic methodology emphasizing the sub-skills expected of the general ILO. The review of the Moroccan curriculum will propose a description of the methodology adopted in 7th grade textbooks for this purpose.

2.2.3 COMMON ERRORS AND DIFFICULTIES

Understanding the common errors and difficulties faced by students in the process of learning the concepts and skills described so far is crucial for any effective assessment and personalized accommodation of students' needs [4]. Educators typically build this knowledge through years of practice and, for the interested, consultation of pedagogical research on the topic. Furthermore, examining errors can reveal a lot about the thinking process and acquired knowledge, with its potential misconceptions, of a given student [4]. When done within the framework of a formalized theory about expected student errors and their interpretations, the error analysis is likely to be more robust, and hence more effective.

For the context of this project, three main sources were consulted for error identification and categorization in early algebra ([2], [3], and [5]). Overall, the takeaways from this literature review were that student errors in the domain of Algebra can be divided into two main categories: calculation errors and conceptual errors. Calculation errors are mistakes in arithmetic operations of numbers, while conceptual errors are likely to signal a lack of mastery or misunderstanding of an algebra-related concept (like eliminating a positive number by adding instead of subtracting it from both sides of the equation). Figure 1 shows a summary of the most common errors found from the study of Payne and Squibb [5]. Another interesting finding of that study is the fact that there seems to be no data-supported distinction between slips and errors steaming from misconception. In as far as this project is concerned, this would mean that the ITS should deal with each of these errors as misconceptions to provide conceptual corrections when needed.

While identifying these errors, both for the purpose of immediate feedback and general learning assessment, can be inaccurate due to a lack of specific data and/or identification methods, the current version of the system will implement the mal-rules⁴ mentioned in Figure 1 as best as possible within the scope of the given time and resources. Future testing of the platform with real students will provide data for further investigating these errors and updating the platform's actions accordingly.

- S1 $M \cdot X = N \rightarrow X = M/N$
- S2 $pat1 + | - M \quad pat2 = pat3 \rightarrow pat1 \quad pat2 = pat3 + | - M$
- S3 $pat1 + | - M \quad pat2 = pat3 \rightarrow pat1 + | - pat2 = pat3 + | - M$
- S4 $pat1 = pat2 + | - Mx \quad pat \ 3 \rightarrow pat1 + | - Mx = pat2 \quad pat3$
- S5 $M (Nx + | - P) \rightarrow M \cdot Nx + | - P$
- S6 $M (Nx + | - P) \rightarrow M \cdot Nx + | - M + | - P$
- S7 $M \cdot X + N \cdot X \rightarrow M \cdot X \cdot N$
- S8 $M \cdot X + N \cdot X \rightarrow M \cdot X + N$
- S9 $M \cdot X + N \cdot X \rightarrow M + X + N + X$
- S10 $M \cdot X \rightarrow M + X$
- S11 $M + N \cdot X = \rightarrow M \cdot N + X =$
- S12 $M + N \cdot X = \rightarrow M + N + X =$
- S13 $M \cdot X + N = \rightarrow M + X + N =$
- S14 $M \cdot X + N = \rightarrow M \cdot X \cdot N =$
- S15 $M \cdot X + N = \rightarrow (M + N) \cdot X =$
- S16 $M \cdot X = N \cdot P \rightarrow X = M$
- S17 $M \cdot X = N \cdot X + P \rightarrow X + X = M + N + P$
- S18 $M \cdot X = N + P \rightarrow M \cdot X = N$
- S19 $M \cdot X = N + P \rightarrow M \cdot X = P$
- S20 $M \cdot X = N \rightarrow X = N$
- S21 $M \cdot X = N \rightarrow X = <N/F> / M$
- S22 $M \cdot X = N \rightarrow X = N / <M/F>$
- S23 $M \cdot (N \cdot X + P) \rightarrow M \cdot X + M \cdot P$
- S24 $2X/2X \rightarrow 0$
- S25 $A \cdot 1/A \rightarrow 0$
- S26 $0 \cdot A \rightarrow A$

Figure 1: Algebra Mal-Rules suggested by Squibb and Payne [5]

2.2.4 MOROCCAN CURRICULUM IN LIGHT OF PEDAGOGICAL LITERATURE

Since this project is meant to be used by the Moroccan audience, it would be all the more beneficial to make it mirror the currently adopted curriculum. This way, the system can be used by 7th grade students to practice their algebra lessons, 6th grade students for an early introduction to algebraic methods, and professors of both levels as an assessment tool of the performance of

⁴ Mal-rules, sometimes referred to as bugs, are the systematic rules by which a student error is produced [5].

their students. The data collected by the system can also inform educational research at a national level for evaluative and reformative purposes.

The review of the Moroccan curriculum in this context was done to assess how it approached the teaching of the concepts and skills discussed in earlier sections, whether this approach covered the necessary pedagogical ground revealed by the literature review, and how the approach could be translated to an ITS curriculum and potentially improved in the process. The resources I used in this process are the 7th grade⁵ textbook “المفيد في الرياضيات” edition 2008 [6], the 6th grade text book “الجيد في الرياضيات” edition 2005 [7], the official curriculum description from the Moroccan Ministry of Education [8], and interviews with two middle school professors in Moroccan private and public schools. Appendix A contains the English version of the interview support document describing the questions asked during the interviews and the style adopted for note-taking.

The major findings of this review are that equations are not introduced until 7th grade, while many pre-algebra techniques are covered in earlier levels, typically in chapters dedicated to problem solving (each year starting 4th grade has at least one lesson titled “Mathematical Problems” or “المسائل” in Arabic). The most notable technique is the one introduced in 6th grade, which uses segments to represent related variables of a problem, and relies on basic manipulations through addition and multiplication to find the final values. Figure 2 shows an example of this method being used to figure out the age of three people, given mathematical relationships between the values of the ages. Figure 3 shows the same method being applied to a problem on CMU’s ITS MathTutor.

⁵ 7th grade and 6th grade corresponds to the first year of middle school and the last year of primary school respectively the Moroccan grade terminology, hence the textbooks used are of those two levels.

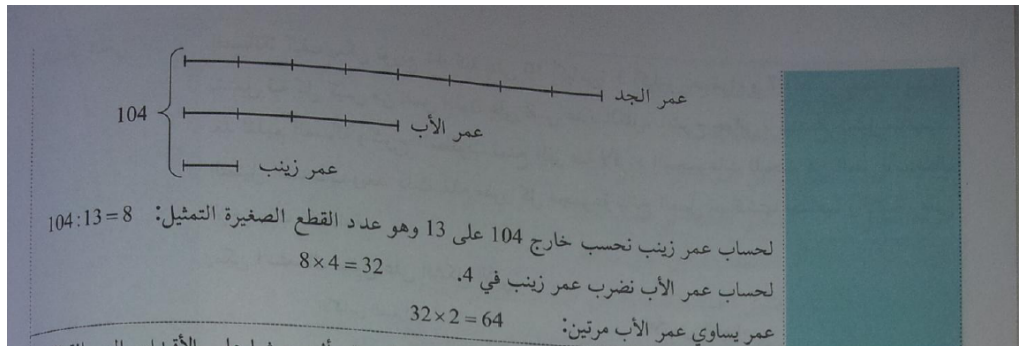


Figure 2: Excerpt from Moroccan Textbook Showcasing the Use of the Segments Method for Problem-Solving

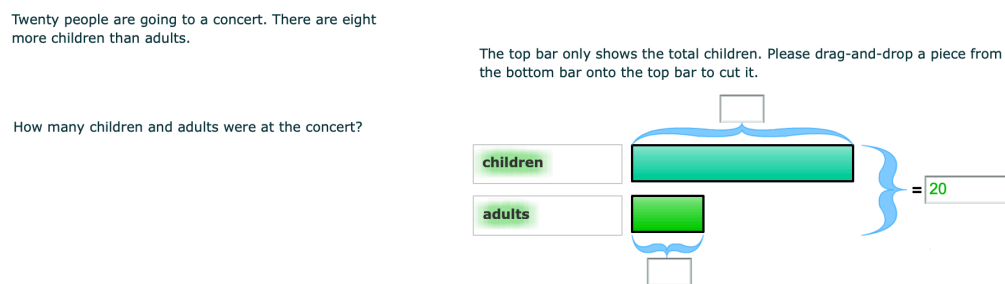


Figure 3: MathTutor Problem-Solving Module Using the Segment Method

Another interesting observation on the Moroccan curriculum is its introduction of equation as a problem-solving tool before they become a formal object of study. In fact, equations are not studied in themselves until 8th grade. Calculations on variables is also implicitly introduced in advanced lessons of arithmetic operations, such as factorization and distribution of an operation on another one, without being the central topic of any lesson in 7th grade. In the meantime, the lesson “Equations” makes use of first-degree equations with one unknown to solve corresponding problems, covering many common algebraic manipulations in the process. The lesson goes as far as to introduce formally two types of first-degree equations: $ax=b$, and $a+x=b$. Exercises following this lesson expect the students to handle relatively complex equations, requiring multiple steps of simplification (including operations on the literal value that is the unknown) before reaching one of the two equations mentioned above.

Interviews with the two professors Mr. Daoudi and Mr. Ibrahim confirmed the observations made on the curriculum. According to professor Daoudi, various informal problem-solving methods are taught to students in primary school before they are replaced by equations starting 7th grade. Mr. Daoudi added that he personally used the segment method to introduce the topic

of first-degree equations, by showing students how one method translates to the other. Mr. Ibrahim, specified that, from his experience, students are rather receptive of basic algebraic knowledge thanks to their pre-algebra training. When asked about their opinion on the key algebra concepts reported in literature, both professors confirmed witnessing some of their 7th and 8th grade students struggle with the algebraic interpretation of the equal sign and literal calculations. Regarding the creation and potential use of the ITS, Pr. Daoudi showed enthusiasm towards the technology, going as far as believing that it can operate without the supervision of a human tutor, while Mr. Ibrahim's opinion was more conservative, since he asserted that human interaction is still paramount in the teaching activity.

The shortcomings of the Moroccan curriculum as seen in light of the literature review include its lack of explicit emphasis on concepts that are known to be problematic in pedagogical research, like the equivalence property of algebraic equality and how it affects the transition to solving equations with unknowns on both sides. Another issue concerns the lack of formal practice with literal calculations, which can make a considerable difference in how students solve equations requiring extensive simplifications. Finally, the textbook did not offer enough material to explain the use of equations and their nature as a primary resource of information on the topic. This is typically true of Moroccan K-12 textbooks, which offer a summary of the main concepts of a lesson and corresponding exercises, and leave the weight of explanation to the professor. In the following section, these observations will be compiled to create the content of the ITS based on the Moroccan curriculum.

2.3 RESULTS: ITS CURRICULUM

Based on the literature review and study of the Moroccan curriculum discussed earlier, the content of the ITS was divided into three modules. This section describes each module with regards to the exercises it contains and the skills they intend to teach.

Module 1: Introduction to First Degree Equations with One Unknown

This module leverages the acquired arithmetic knowledge of students to introduce first degree equations as a methodic tool to approach problems students are already familiar with. The problems chosen for this module are known in literature as 'Word Equations' [3], since they express in words a relationship between a set of variables without any real-life context. An example of such problems would be: *"If you subtract 3 from a number and you get one, what*

number did you have initially?”. In addition to being recognized in pedagogical research for their simplicity, these problems are used in the primary school Moroccan curriculum to introduce subtraction and division as reverses of addition and multiplication [8]. This module will hence contain a set of word equation problems and force the student to solve them using formal equations instead of the pre-algebra methods they are familiar with. The aims of this module, alongside the strategies implemented to achieve them and the expected students’ difficulties for each one of them, are detailed in Table 2.

Compared to the Moroccan curriculum, this module adopts the same approach of introducing equations as a problem-solving tool, with the added value of following the recommendations of literature in terms of concepts to focus on and errors to expect. It is worth noting that each of the skills mentioned in Table 2 can make the subject of a single and/or set of problems. The problems of this module will all share the same interface as described later in this report.

Module 2: Solving Equations

This module is perhaps the most straight-forward of all. Students will be given one equation per exercise and asked to solve it and report their result at the end. In this process, they would also be required to explicitly state the type of ‘action’ they performed at each step of the resolution process, to make sure correct results are not a coincidence. The interface for this type of problems will be as flexible as possible to allow a full range of actions, and hence mistakes, to the student.

Module 3: Using Equations to Model and Solve Problems

This module is a direct implementation of the algebraic problem-solving methodology proposed in the Moroccan curriculum. This methodology contains five distinct steps for solving any one problem: 1) expressing the problem variables using a single unknown⁶, 2) generating the mathematical equation best describing the problem, 3) solving the equation to find the value of the unknown, 4) computing the final values of the initial variables, and 5) double checking the results using the conditions of the problem. This module will hence implement a few exercises from the ones proposed in 7th grade textbook to allow students to practice problem-solving following this methodology.

⁶ The problems tackled in 7th grade all require solving a single equation with one unknown. Therefore, even when they contain multiple variable, they should all require finding a single value.

Table 2: Module 1 Sub-Skills, Strategies and Expected Difficulties

Goal	Strategy	Expected Difficulties
Introducing notion of the unknown	<p>The missing number of the problem statement will be explicitly named 'x' before translating the entire statement to a mathematical expression.</p> <p>The expression will be translated as is, in the order in which the values appear in the text. This would keep the introduction of the notation 'x' as the only novelty for this type of problems.</p>	None. The unknown can also be given a more meaningful name if it is believed to help the understanding of students.
Check mastery of arithmetic skills	<p>The equations will require a minimal number of simplifications and computations in the process of finding the value of x. The numbers and operations involved can increase in difficulty to check that the student has a strong enough background in arithmetic to support his/her study of Algebra.</p> <p>At this stage, the order of the terms of the equation on one side might be reversed to check for the assimilation of arithmetic rules such as operation precedence, transitivity, etc.</p>	Accumulated difficulties in arithmetic manipulations. This is a likely indicator on the type of mistakes to expect from the student at a later stage, and an assessment of whether or not they are ready to study algebra.
Equality as an equivalence	This can be achieved by reverting the left- and right-hand sides of the generated equation or one of its equivalents in the resolution process.	None, except the novelty of the situation.
Introducing notions of algebraic manipulations	This is achievable in the process of solving the given equation. The student will be forced to follow a pattern imposed by the system to ensure that his solution does not rely on informal pre-algebra methods.	Conceptual errors such as: <ul style="list-style-type: none">- Performing an operation on one side of the equation and not the other.- Performing the wrong operation in the process of removing a term.- The list of mal-rules in Figure 1.

3 TECHNICAL RESEARCH

3.1 INTRODUCTION

Intelligent Tutoring Systems are almost as old as artificial intelligence itself [1]. In fact, automating instructions can be traced back to 1926, with the mechanical multiple-choice system developed by Pressey, while intelligent instructional systems started emerging in the 1970s [9]. What began as an idea to reduce the cost and effort of human teaching through the involvement of machines soon developed to the goal of renovating education through adaptive and/or personalized learning environments. If machines have the potential to automate content delivery, process the generated learner data for assessment, and sequence the instructions delivered based on a set of criteria, then they seem to have the basic ingredients for creating learning environments tailored to meet the needs of every individual [9]. Years of research have proven that this endeavor is not as simple as it appears to be. Despite the recent explosion in data and data exploration techniques (including the fields of Data Science, Big Data, and Machine Learning, to only name the most famous ones), personalizing education is still not achieved in its fullest conception.

That being said, a lot has been achieved in this domain over the past decades. If the goal is not reached yet, it surely is not from a lack of trials with positive outcomes. Many techniques for adapting learning environments and/or improving the effectiveness of teaching in general have been investigated both in theory and in practice [1]. This section is dedicated to exploring behind-the-scenes work of adaptive educational technology, particularly Intelligent Tutoring Systems, and highlighting the major achievements of the field. In the coming paragraphs, I will describe two types of ITS: cognitive tutors and statistical modeling tutors. I will also provide an overview of the most recent advances in the field, the topics they aim to explore, and the techniques/technologies they use in the process. Finally, I will report on a specific ITS which I have used as my main inspiration for this project (CMU's MathTutor), then present an overview of the technical design of my system.

3.2 LITERATURE REVIEW

3.2.1 *TRADITIONAL ARCHITECTURE OF INTELLIGENT TUTORING SYSTEMS*

One of the most widely used architectures for an ITS divides the system into four components: domain modeling, student modeling, tutor modeling and user-interface [18]. This architecture is also known as the 'traditional trinity', in which case the user-interface is considered part of

the tutor modeling. The idea behind this architecture is to identify the parts of an ITS requiring knowledge and reasoning, leaving the matter of their integration to be investigated through various strategies [13]. The conception and development of this project can be understood within the framework of this architecture. Therefore, a description of each of these components is owed in order to standardized the conception and development of this system for future work and referencing.

The first component of the ITS architecture, known as domain modeling or expert knowledge, is concerned with the field in which the tutor is operating. Work in this module aims to analyze the domain to be learned in terms of concepts, rules, and problem-solving strategies [13]. The analysis in this regard is twofold: defining the curriculum (content and content sequencing) of the domain and choosing a way to represent it within the system. Various methods have been investigated to formalize the acquisition and representation of knowledge domain in tutoring systems. In this project, knowledge acquisition was done through the pedagogical research, collecting information through literature review and discussions with domain professionals, while its representation was handled by the technology chosen for implementation in the form of a production system.

The second component of the system looks at the modeling of the student's cognitive and affective states within the system. This model is considered by many to be at the core of the ITS, since it allows the tracking of students' progress and analysis of their learning behavior in order to implement system adaptability and individualization of instructions [13]. In fact, student modeling is such a crucial and challenging topic that it is considered many times a field of study of its own. In its essence, a student model serves three main functions: 1) gather data about the learner, 2) use the collected data to represent the student's knowledge and learning process, and 3) analyze the generated model to inform content sequencing and progression evaluation of the learner [18]. Sections 3.2.2 and 3.2.3 will each present an overview of one method for implementing and using student models.

Tutor modeling refers to the sum of decisions taken by the tutor based on input from student and domain models. Such decisions include the form and timing of intervention in addition to content sequencing and tutoring strategies [18]. This component defines the behavior of the tutor, and must rely on artificially intelligent analysis and data collection to make decisions in

the best interest of the learner. For this project, this module was handled by the authoring tool chosen for creating the ITS, through a set of built-in functions discussed in section 5.3.

Finally, user-interface is the connecting point between a tutor and a learner. It serves to collect the user's input as well as render the actions of the tutor. It may come in different forms and media, including a web/mobile app interface, virtual-reality environments, simulations, etc. The system developed in this project interfaces through a web application.

3.2.2 STUDENT MODELING: MODEL TRACING TUTORS

Model tracing tutors are a type of Intelligent Tutoring Systems implementing a cognitive theory to generate a model of an ideal cognitive schema against which the student's behavior is compared during the instruction process [1]. This enables tracing the cognitive skills and knowledge of a student to assess their performance and decide on a course of action that is likely to improve their learning (i.e., getting them closer to the pre-defined cognitive model). This tutors' model requires significant knowledge about the domain of instructions, to be potentially knowledge engineered if the implementation of the system relies on production-rules [10]. Another requirement of this type of ITS is a cognitive theory/framework within which the ideal behavioral model is implemented. One of the most-known such theories is ACT-R, set forward by Anderson in 1987 [1].

3.2.3 STUDENT MODELING: KNOWLEDGE TRACING TUTORS

Unlike their cognitive counterparts, knowledge tracing tutors do not rely on a pre-defined model or a set of theoretical assumptions on human behavior. Instead, they make use of statistics to process a learner's behavior and adapt their actions accordingly [1]. These systems require an extensive amount of learners' data to find statistically viable trends and patterns to inform the implementation of the tutor. They also need a specific statistical method to harness their data and frame their assumptions. One of the most typically used tools for this kind of systems is Markov processes for statistical analysis [1].

3.2.4 RECENT ADVANCES IN ITSs

ITSs were not immune to the enthusiasm around machine learning [11]. Over the past decade, more and more systems aim to incorporate advanced data mining techniques into their toolkit. One challenge of using generic data analysis tools in a domain as complex and multi-faceted as

education is accommodating the unstructured nature of the field, coupled with the challenging task of defining the ILOs of every knowledge/instruction domain. In the foreseeable future, even the most enthusiastic machine learning supporters can hardly argue for the obliteration of the human factor in designing educational technology.

Aside from an upgrade of the techniques used in the field, the topics tackled in conferences about ITSs and related topics are more and more specific and ambitious. A recent trend in ITS development for instance seeks to endow the systems with emotional intelligence, so that they can adapt to the emotional state of a learner in addition to their displayed cognitive behavior [12]. The emotions of a student are captured through physical sensors and interpretation of the student's interaction with the virtual system. This data influences the type of feedback and content sequencing the system provides.

3.3 RESULTS: TECHNICAL CHOICES FOR THE ITS

Based on this literature review, I have decided to make the ITS of this project a model tracing tutor. The context of development of this project, including its limited time frame and the lack of data resources needed for its creation contributed majorly to this decision. Another factor is my relative familiarity with cognitive theories through previous school work and overall fascination with the field of cognitive science and cognitive modeling. A third element influencing my decision is the prospect of using rule-based production systems and declarative programming, which I was briefly introduced to in the course of Languages and Compilers but never truly worked with since. Finally, a last motive for choosing model tracing tutors is the availability of CMU-developed tools for the creation of such systems, which will be further discussed in the following section.

4 OVERVIEW OF CTAT

4.1 INTRODUCTION

CTAT stands for Cognitive Tutor Authoring Tool. As its name hints, this software is a comprehensive framework for creating cognitive tutors of the model tracing type, from interface implementation to the writing and execution of the system's production rules [14]. Following the conceptual design of model tracing tutors defined earlier, we know that an ITS of this type relies on a student model that it builds within the framework of a cognitive theory. For CTAT, this theory is ACT-R [14]. As such, the software requires encoding the knowledge domain of the ITS in the form of skills acquired sequentially. The implementation of this knowledge domain is done using a rule-based production system implemented in Jess [14]. CTAT provides a number of supporting tools for the authoring of the production system, its execution and debugging. For non-programmers, first-timers, and people only interested in proto-typing, CMU's authoring framework also offers an XML graph-based type of tutors, called example-tracing tutors [15]. The following sub-sections contain a presentation of the main tools and features of CTAT, an overview of the Jess language and declarative programming in general, and a comparative analysis between example-tracing and cognitive tutors. The chapter concludes with a description of the implementation plan of this ITS using CTAT.

4.2 COMPONENTS OF CTAT

4.2.1 HTML EDITOR

The first step in developing a cognitive tutor, after downloading, installing and connecting the software with Google Drive or DropBox as per the instructions in the documentation, is creating the system's interface [14]. To this aim, CTAT offers an interface building tool called HTML Editor. This web-based application allows the creation of customized user interfaces through dragging and dropping CTAT built-in elements, which are translated to HTML components with their corresponding Javascript and CSS files [13]. Figure 4 shows a screenshot of the interface of the HTML editor on login, while figure 5 shows the process of adding front-end components to the initial background element.

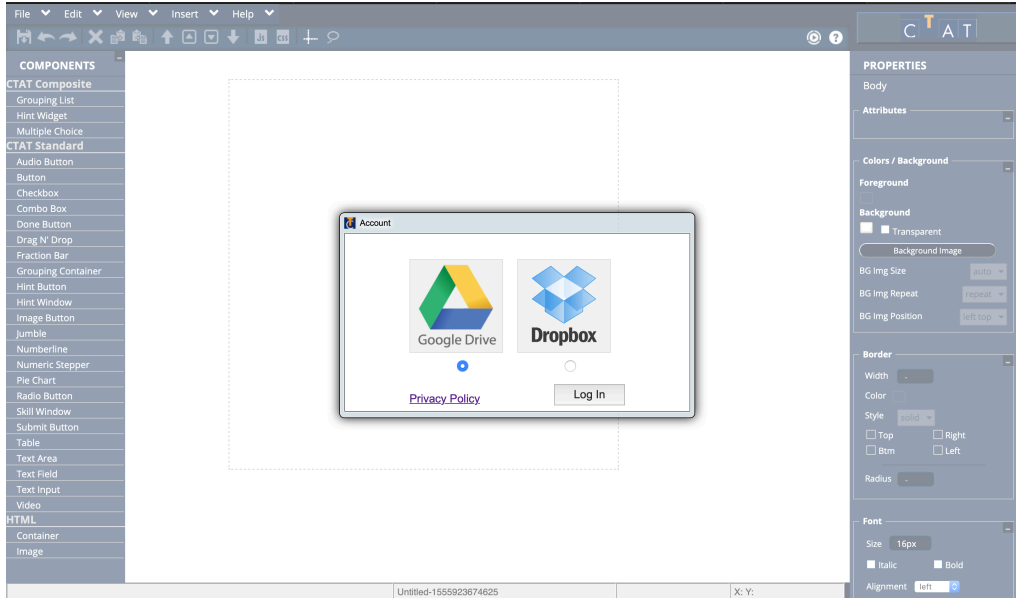


Figure 4: HTML Editor Login Page

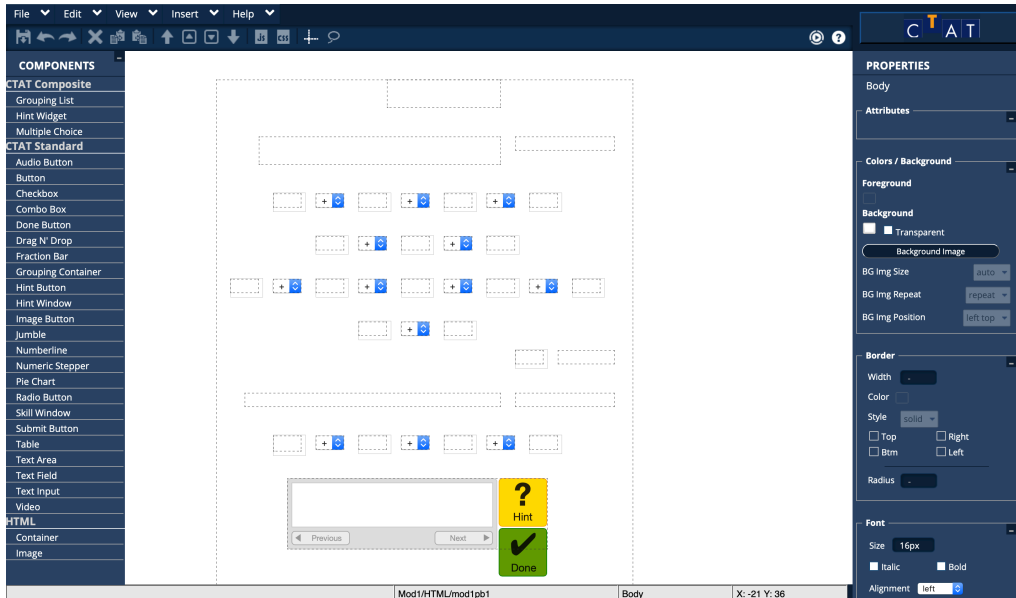


Figure 5: HTML Editor in the Process of Creating a Student Interface

The editor saves every file it creates inside a package located on whichever cloud storage option (Google Drive or DropBox) was chosen during the configuration of CTAT [13]. Each package follows the structure shown in Figure 6. The HTML folder contains interfaces of the problems presented in the tutor. The FinalBRD folder holds the graphs run by the said problems (more on this in the chapter describing the final implementation of the ITS), while the CognitiveModel folder is home to the files required by the rule-based production system embedded in CTAT.

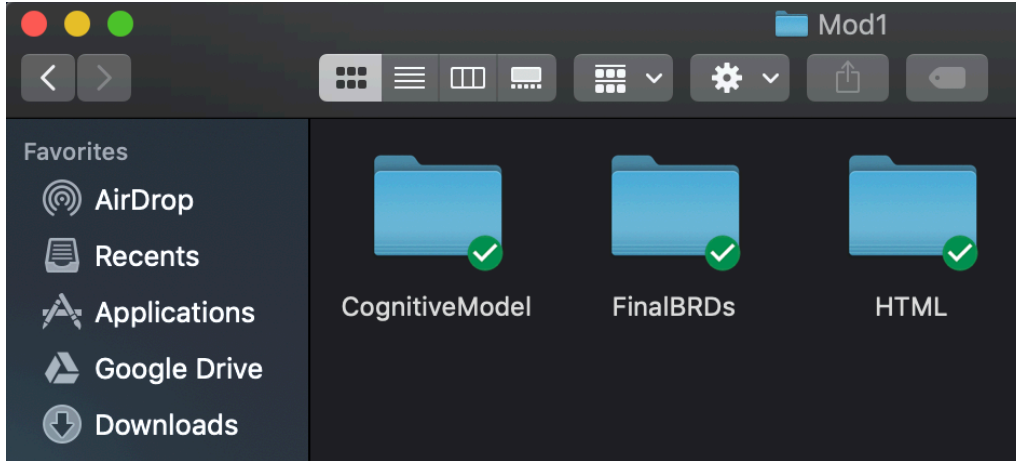


Figure 6: Folder Structure of a CTAT Package

An exercise in the tutor is a coupling between a student interface and a behavior graph. Therefore, a single interface can be used by many problems, the data of which can vary depending on the paired graph. When using the HTML editor for interface building, every interface would have two affiliated HTML files, one ending in *.ed.html* and the other one in *.html* [13]. The first extension belongs to the file created by the editor, while the second one is the official HTML file which is launched by the tutor during testing and deployment, and augmented during development to support capabilities not offered by the editor, such as dynamic generation or resizing of components. A good way to view the editor is as a tool to generate the blueprint of the interface, which can then be customized manually through the *.html* file [13]. Once the interface is ready, it can be launched through the CTAT HTML and Flash software, which is used to develop and test cognitive and example-tracing tutors alike. The next section provides an overview of this tool.

4.2.2 CTAT FOR HTML AND FLASH

This software is used for developing the back-end of tutors whose interface is built using HTML and Flash⁷ [14]. The software offers the option of rendering any interface developed for the tutor in a browser window by running its *.html* file. As soon as the interface appears on the screen, the system launches the behavior recorder, whose job is to keep track of the actions performed on the student interface to be labeled and analyzed by the developer of the system to form a cognitive model for a specific problem. For the behavior recorder to be running, the Author Mode option should be set to *Demonstrate* as shown in Figure 7. Other values for this

⁷ CTAT offers another software for tutors with a Java-powered interface.

feature include *Test Tutor*, for testing the model without tracking more action, and *Set Start State*, for defining the initial state of the behavior graph.

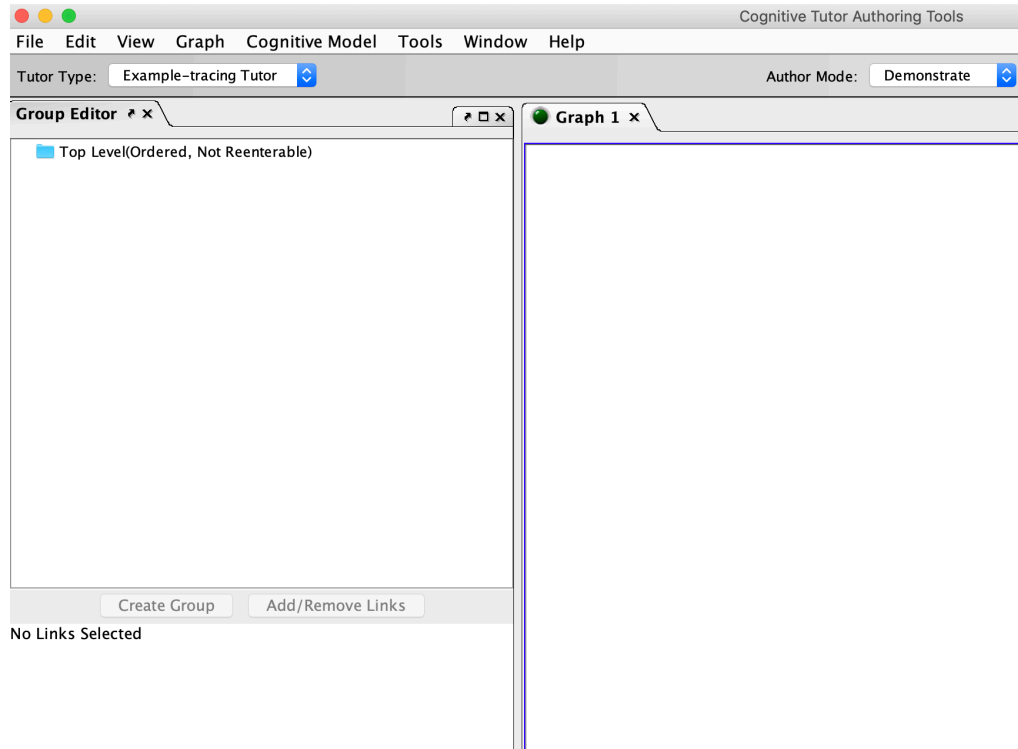


Figure 7: CTAT HTML and Flash Software Set to Demonstrate Author Mode

Depending on the type of tutor envisioned, CTAT offers a number of support tools to simplify the planning, creation and testing of the final system. For example-tracing tutors, the most interesting of these features is the behavior graph (see Figure 8 for an example). This tool allows representing snapshots of the state of the interface as nodes of the graph, with the student actions leading to those states as links between consecutive nodes [17]. The software allows creating new states manually (for tutor performed actions which sometimes update the interface under certain conditions, by hiding/showing some interface components for instance), modifying the information contained in an existing state, and most importantly, labeling the actions on the links with the corresponding skill they showcase. For cognitive tutors, this software offers a set of tools for creating the rules of the system, tracking their execution, and resolving any bug they might generate [16]. These tools will be visited in greater detail in the following section, while reviewing the difference between example-tracing and cognitive tutors and the use of each.

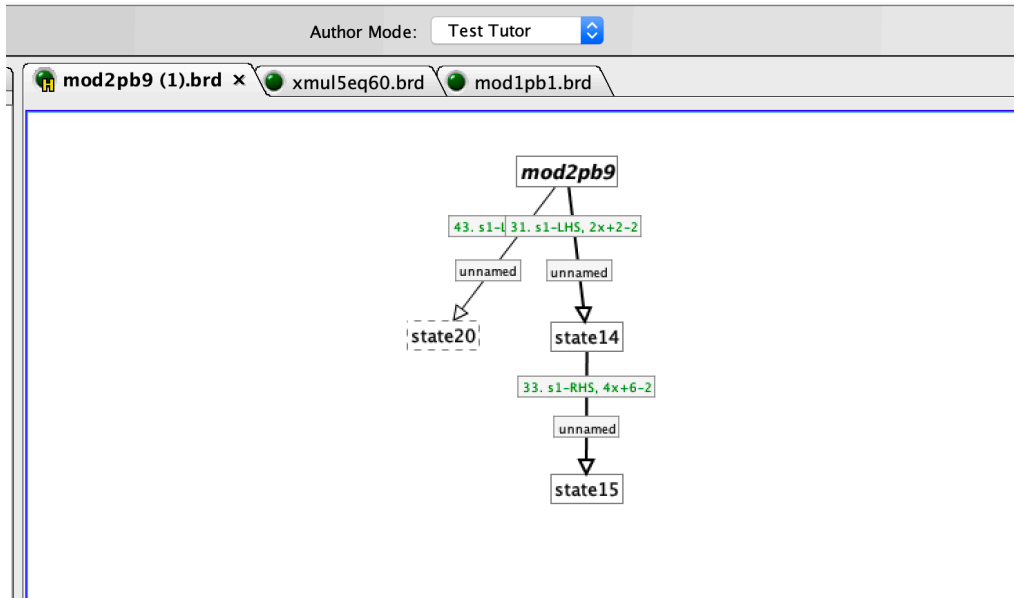


Figure 8: Example of a Behavior Graph

4.2.3 EXAMPLE-TRACING VS COGNITIVE TUTOR

Example-tracing tutors were CTAT's solution to tutor authors who are not familiar with programming [15]. The creation of these tutors can be done entirely through the interface of either software products: *CTAT for HTML and Flash* or *CTAT for Java*. To create an example-tracing tutor, the author needs to run the student interface of a given problem, set the 'Author Mode' to *Demonstrate*, and solve the problem manually through the interface. To show alternative correct solutions, the author can reset the state of the problem to any of the ones saved in the initial recording, and trace the alternative branch from its starting point. Wrong actions can also be demonstrated in a similar manner and marked as bugs on the problem's graph. Finally, feedback and assessment can be added to the graph's nodes in the same way in which skill labels are added to action links [15].

Aside from being a code-free alternative to the creation of intelligent educational environments, example-tracing tutors can help in designing and testing cognitive tutors. Before deciding on which skills to code-up as rules and which errors as mal-rules, a tutor author can run a few tests on the interface and use the behavior recorder to record, sort and label their actions as a form of knowledge engineering of the domain [15]. After creating the rules of the system, saved behavior graphs can serve as test cases to check if the action modeled through rules and facts are correct. Another potential use of example-tracing tutors is prototyping a system as a proof of concept and/or to collect data from a real-life audience before coding up the rules necessary

for cognitive tutoring [15]. For all of the reasons mentioned above, an example-tracing tutor was developed for every module of the ones contained in this ITS (more on this in chapter 6).

CTAT cognitive tutors, as explained earlier, rely on rules and facts written in Jess and making up the knowledge base of the system [16]. Cognitive tutors have the advantage of being robust and scalable compared to the example-tracing ones, since they use rules written and modifiable by the creators of the tutor at any given time [16]. Creating the brain of a cognitive tutor requires extensive expertise in production systems and declarative programming (more on these in the following section). From defining the facts of a problem and their corresponding templates, to developing rules to mimic student actions through declarative programming, the processing power of cognitive tutors requires a strong understanding of the domain in which the tutor is operating, a good grasp on the concepts of rule-based programming, and mostly a lot of patience given the iterative nature of the process. Luckily, CTAT has a set of tools to assist with the development of rule-based cognitive models.

The first tool provided by CTAT for rule developers is a Jess console, shown in Figure 9. In addition to giving real-time feedback about the loading and syntax errors of the system's files, the console allows interacting with the rules of the system by running general Jess commands through the console input field. For the developers unfamiliar with Jess syntax, or preferring the comfort of a built-in interface, CTAT offers Working Memory Editor window, shown in Figure 10, which displays the content of the working memory of the system and offers the option to modify it and add on to it through buttons and fields [16]. To keep an eye on the execution (or firing) of production rules, there is an agenda window displaying in real-time the forward-chaining process used by Jess to match the LHS of a rule using facts in working memory and user input. Finally, the 'Why Not' window, launched by clicking on a particular rule from the execution tree, provides details on the rules tried during the matching process for the developer to assess the rules behavior [16].

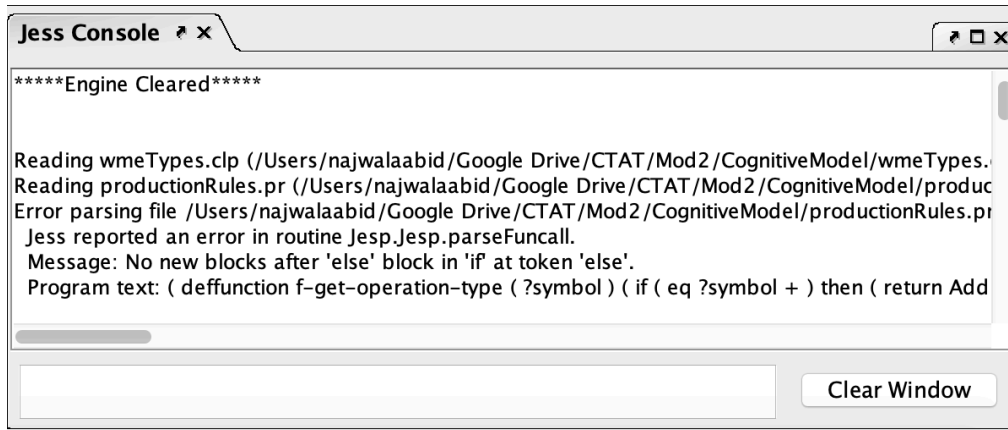


Figure 9: Jess Console on CTAT for HTML and Flash

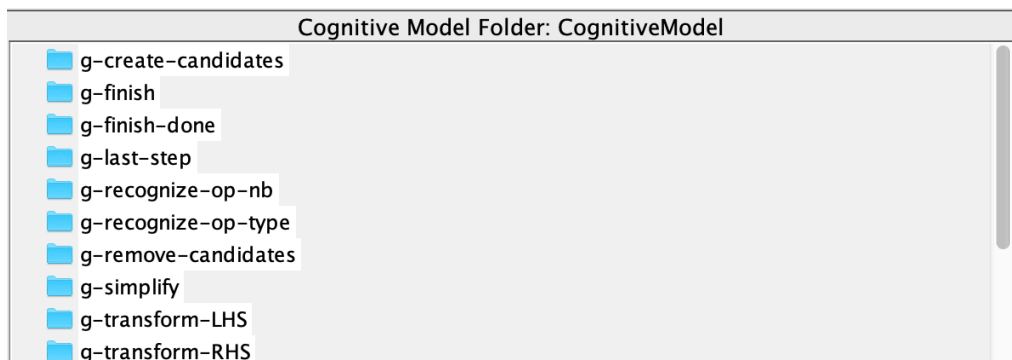


Figure 10: Working Memory Editor Window from CTAT for HTML and Flash

4.2.4 JESS LANGUAGE AND DECLARATIVE PROGRAMMING

Jess is a Java-based variation of the functional language LISP. It is commonly used for creating production systems due to its ease of integration with Java, which expands considerably its features and hence scope of use. Despite its relatively wide user base, because of the constrained nature of production rule systems, the documentation available on Jess is minimal. There are a few user books, one of which was a main resource for this project, and one main website, with a rather archaic navigation system and user interface. This made working with Jess a real challenge, especially for a first-time user like myself.

Jess in itself, regardless of its strong features and suitability for production systems, remains a single entry-point to the wide world of declarative programming. Declarative programming is a paradigm in which the programmer defines facts and rules and lets the system figure their way around them using an inference engine. This is starkly different from procedural programming, perhaps the most common form of all-purpose programming, in which a

developer needs to provide an algorithm of discrete consecutive steps for a system to execute. In the procedural paradigm, the developer provides the *how* of a solution, while declarative programming specifies the *what* and lets the inference engine do the rest.

Thinking declaratively was a real challenging for a procedurally trained programmer like myself. Wrapping my head around concepts of facts and rules, when my brain was screaming solutions involving graphs and other procedural style data structures, was akin to swimming against current in unknown waters. Luckily, I managed to develop the rules behind the modules of the system, as described in chapter 6.

4.3 RESULTS: TECHNICAL DESIGN OF THE SYSTEM

As mentioned earlier, the implementation of the system made use of both example-tracing tutors and cognitive tutors, each within their recommended scope of use. Example-tracing tutors allowed me to recognize the skills required by each module, in greater detail than the pedagogical research. Translating those skills to rules of the system with accompanying facts for each specific problem came at a later stage, and was mostly an iterative process involving a lot of testing of each rule using the tools provided by CTAT.

5 ITS: SYSTEM DESCRIPTION

5.1 INTRODUCTION

After discussing the findings of the pedagogical and technical researches and providing an overview of the technology used in implementation, it is time to report on the final product. This chapter will provide a description of the interfaces used by the system, and the behavioral graphs and rules developed for each module. Since the system is made of three modules as per the proposed curriculum (see chapter 2), each of the components discussed below will be looked at within each module separately.

This part of the report is mainly a description of the solution proposed as a synthesis of all the work reported so far. It is therefore primarily personal work, with little use of sources of inspiration/external resources. The sources and resources will be credited whenever present.

5.2 OVERVIEW OF THE SYSTEM COMPONENTS

5.2.1 STUDENT INTERFACE

Module 1: Introduction to Equations

The first module aims to introduce students to basic concepts relevant to the study and use of first-degree equations with one unknown, as the introductory topic of early algebra. The module assumes that the learners were never exposed before to this topic. It is meant to be used by 7th grade students at the beginning of the school year, or 6th graders at the end of their curriculum as means to get accustomed to the topics covered the next year. The system can also serve as a stand-alone introduction to early algebra, although this claim is yet to be supported by data about the students' reaction to the system and its overall performance with a real-life audience.

Since the module is designed as a first exposure to the concepts of unknown, algebraic manipulations and the equivalence meaning of algebraic equality, it implements a rather constrained student interface leaving little freedom of action for the students' input. Four levels of difficulty of problems are suggested to reflect the four major skills intended to be covered by this module (see chapter 2 for more details). Appendix B shows the interface and initial state of each of those variants of problems. It is worth noting that the example reported in the appendix uses the same values but different arrangements of the student interface. In reality, each of those problem levels would make use of its own set of values as long as the overall

problem type remains the same: simple word equations with one unknown and a series of arithmetic terms (e.g., pb1: $x+2-3 = 12$, pb2: $3*x+5=20$, etc.).

Module 2: Solving Equations

The interface of this module was inspired by that of MathTutor, a CMU powered ITS for tutoring algebra. The interesting facts about this interface are that it allows students a maximum flexibility in their input, and requires an explicit labeling of the action taken at each resolution step. Figure 11 shows this interface in action. The last step of the solution makes the student restate the final result to make sure they understand what they are doing.

الدرس الثاني

حل المعادلة الآتية:

$2x+2$	=	$4x+6$	
$2x+2-2$	=	$4x+6-2$	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px 5px;">طرح عدد من طرفي المعادلة</div> <div style="border: 1px solid #ccc; width: 30px; text-align: center; margin-left: 5px;">2</div> </div>
$2x$	=	$4x+4$	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px 5px;">تبسيط طرفي المعادلة بعدد</div> <div style="border: 1px solid #ccc; width: 30px; height: 15px; margin-left: 5px;"></div> </div>
	=		

=

الحل هو:

?
Hint

✓
Done

◀ Previous

Next ▶

Figure 11: Module 2 Student Interface

Module 3: Problem-solving

Since this module aims at implementing directly the five-step problem-solving methodology proposed in the Moroccan curriculum, the interface was kept minimal to reflect this intention as shown in Figures 12 and 13. The downside of this implementation is that it is not easily

scalable to other kinds of problems. Implementing scalability requires a deeper analysis of the types of problems proposed by the curriculum and how to abstract them to a common interface. This investigation will be carried out in future work.

الدرس الثالث

المسألة: أوجد مثلثا محيطه 30 مترا و أطوال أضلاعه ثلاثة أعداد صحيحة متتالية.

اختيار المجهول: ليكن x طول أصغر ضلع للمثلث. عبر عن القيم الآتية بدلالة x .
الأضلاع 1 إلى 3 مرتبة من الأصغر إلى الأكبر.

ضلع 1 ضلع 2 ضلع 3

صياغة المسألة: حول الجملة التالية إلى تعبير رياضي: "مجموع الأضلاع الثلاث يساوي 30".

=

حل المعادلة: اكتب المعادلة على شكل مختزل:

=

=

=

التحقق من شروط المسألة: تأكد أن القيم المحصل عليها تحقق شروط المسألة.

+ + =

Figure 12: Module 3 Student Interface (1)

الرجوع إلى المسألة: ضع القيم المحصل عليها في المكان المناسب:

ضلع 1 ضلع 2 ضلع 3

التحقق من شروط المسألة: تأكد أن القيم المحصل عليها تحقق شروط المسألة.

+ + =

Figure 13: Module 3 Student Interface (2)

5.2.2 GRAPHS AND SKILLS

As mentioned in the previous discussion about example-tracing and cognitive tutors, graphs are proposed by CTAT as a tool to visualize the actions captured by the behavior recorder for analysis and labeling. Figure 14 shows an example of such graphs, while Table 3 summarizes the skills captured within every module. Since each of the modules proposed in this ITS

contains an example-tracing and a cognitive tutor, they each include at least a single XML file containing data about the behavioral graph.

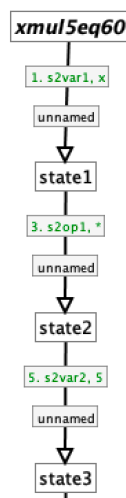


Figure 14: Sample Graph from Module 2

The behavioral graph in Figure 14 traces the first steps in solving the word equation ‘*we multiplied a number by 5 to get 60, what was the initial number?*’. The steps shown on the graph capture the student input (x in field s2var1, *⁸ in variable s2op1, etc.). State 1, state2, and state 3 are each a snapshot of all three models of the system (student, domain, and tutor) after each input. Clicking on any of these states will update all of the models to reflect the values associated with that state.

5.2.3 PRODUCTION RULES

The skills mentioned in section 5.2.2 are translated to facts and rules in the Jess language to build a cognitive model of the expected student behavior within each exercise. These production rules vary widely from a skill to another, and were typically implemented through analysis of the skill and trial and error in the Jess language. Most of them rely on helper functions implemented locally as part of the production system. As an example, Figure 15 shows a rule for recognizing the operation performed at a single resolution step from Module 2.

⁸ Multiplication sign.

```

(defrule r-recognize-op-type
  ?pb <- (problem (equation ?LHS ?RHS) (subgoals $? ?subgoal $?) )

  ?subgoal <- (g-recognize-op-type (value ?v) (field ?f) )

  =>

  (retract ?subgoal)
  (predict-observable-action ?f UpdateComboBox ?v)
)

(defrule r-recognize-op-nb
  ?pb <- (problem (equation ?LHS ?RHS) (subgoals $? ?subgoal $?) )

  ?subgoal <- (g-recognize-op-nb (number ?nb) (field ?f) )

  =>

  (retract ?subgoal)

  (predict-observable-action ?f UpdateTextField ?nb)

  ; get next-field
  (bind ?next-f (f-get-next-field ?pb) )

  ; add subgoal to simplify LHS
  (bind ?sub (assert(MAIN::g-simplify (side-type LHS) (field ?next-f) ) ) )

  (printout t crlf "==== next-field: " ?next-f crlf)

  (modify ?pb (subgoals ?sub) )
)

```

Figure 15: Sample Rule from Module 2

Figure 5 shows two rules: *r-recognize-op-type* and *r-recognize-op-nb*. The first rule is used to evaluate student's input for labeling the operation performed on a step of the resolution process, whereas the second one is used to capture the number or term involved in the operation. For instance, subtracting 2 from the two sides of the equation $2x+2=4x+6$ translates to the operation type '*subtraction*' with the number value '2'. Every rule is divided in two parts: left-hand side, occurring before the arrow sign ' \Rightarrow ', and representing a set of conditions to be fulfilled by facts of the working memory, and a right-hand side, occurring after the arrow sign, and providing actions to perform if the conditions of the rule are all satisfied. The pseudocodes for the two rules are given below.

Pseudo-code for the rule '*r-recognize-op*':

IF:

- There is a problem in working memory with two variables for the slot equation and a list of sub-goals
- If among the sub-goals of the problems found above, there is one for recognizing the given value in the given field.

THEN:

- Remove the goal matched
- Predict the student action of updating the corresponding field with the correct operation label

Pseudo-code for the rule '*r-recognize-op-nb*':

IF:

- There is a problem in working memory with two variables for the slot equation and a list of sub-goals.
- If among the sub-goals of the problems found above, there is one for recognizing the given value in the given field.

THEN:

- Remove the goal matched.
- Update the value of the field where the next action is expected.
- Add a new sub-goal for simplifying the left-hand side of the equation as the next expected solution step.

Table 3: Skills Captured by the Behavior Recorder per Module.

Module	Labeled Skills
M.1: Introduction	<ul style="list-style-type: none">- Recognize x as a reference to the unknown quantity in the problem statement.- Simplify both sides of the equation by performing all arithmetic operations available until the expression is as concise as possible.- Add/Subtract one term/value from both sides of the equation to isolate the unknown.- Report the final result of the equation and click on done.⁹
M.2: Equation Resolution	<ul style="list-style-type: none">- Perform a ‘simplification’ or ‘transformation’ operation depending on the state of the equation. Simplification means combining like terms using the operands linking them, while transformation means generating an equivalent system by performing the same operation on both sides of the equation.- Performing actions that move forward toward solving the equation (transformations which do not get the equation closer to isolating the unknown and redundant steps are not accepted as valid input, despite their mathematical correctness).- Finish the resolution schema when the coefficient of the unknown term is 1.- Report the result and click on the done button.
M.3.: Problem Solving	<ul style="list-style-type: none">- Express the problem’s variables in terms of a single unknown entity.- Find the mathematical equation best describing the relationships evoked in the problem statement.- Solve the mathematical equation mentioned above using the skills from M.2.- Explicitly report on the final values of the problem variables.- Double check the values found against the conditions of the problem statement.- Click on the done button.

5.3 DEPLOYMENT AND TESTING

I chose to deploy the system on CMU’s TutorShop, a free-access platform dedicated to hosting ITSs, and providing a number of tools for managing and expanding their data. In order to use the platform, I had to request access permission from TutorShop’s team. The team responded

⁹ Recognize that the problem is ‘done’ is an important skill since it conveys an understanding of the context of actions of the student.

in a timely manner and gave me full access to upload, use and update my tutor. Figure 16 shows the email granting me access to the platform, while Figure 17 is a screenshot of the home page of the website, and Figure 18 shows one of the exercises of the ITS after its deployment.

The first step in hosting a tutor is to create a .zip file of all its modules. The file is then uploaded to the system from a dedicated tab. The next step is to define a number of problem-set(s), each containing a series of exercises of a similar type. I created a problem-set for every module of the system. The third and last step is configuring the ITS to be used by TutorShop. Multiple settings can be defined for every uploaded tutor, including the problem sequencing algorithm and skills for every problem set. I have chosen problem sequencing to be based on skill mastery, and defined the skills to match the Jess rules of every module.

In addition to hosting, TutorShop offers interesting tools for gathering and analyzing learners' data on the system, through a data management platform called DataShop. DataShop provides features to visualize learners' data for research and modeling purposes. The use of such features for bettering the performance of the ITS will be investigated in future work, after testing the platform with a real audience.

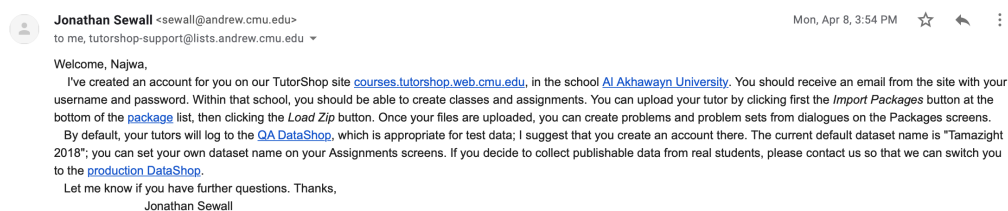


Figure 16: Email Granting Permission to Use TutorShop

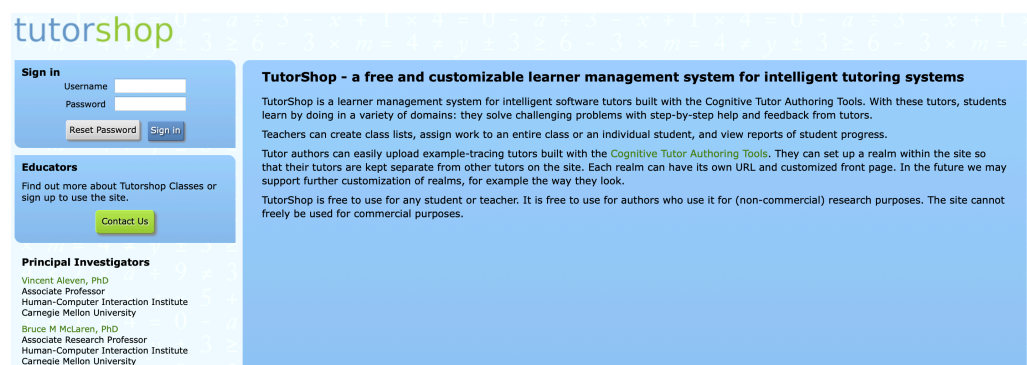


Figure 17: Welcome Page of TutorShop

الدرس الثاني

$$2x+2 = 4x+6$$
$$\square = \square$$

حل المعادلة الآتية:

$$\square = \square$$

الحل هو:

Previous Next

?

Hint

✓

Done

Figure 18: First Problem of Module 2 Hosted on TutorShop

6 STEEPLE ANALYSIS

A STEEPLE analysis of this project is necessary because of the multi-facet implications it entails on society as a whole. Following is a summary of said implications looked at from the perspective of the seven components of the STEEPLE analysis:

1. Socio-cultural

Some of the social and cultural factors that might affect this project are:

- **Target population's level of education and digital literacy:** children may not have the expected supervision from their environment due to the parents' lack of education or unfamiliarity with computer systems, particularly in rural areas.
- **The language of instruction and cultural relevance of the first version of the system:** The system's interface will show text solely in Arabic, including instructions, hints, and feedback. The implementation is also flexible enough to allow for an easy support of additional and/or other languages of instruction in the future.

2. Technological

The technological effects of the environment on the system would involve any limitations in the fields of student modeling and system's adaptability. Given the relatively small scale of the system, the impact of such limitations shall be curbed. The availability of the hardware supporting the ITS is also a concern relevant to this section. To overcome this issue, the final product will be available in the form of a web-application requiring minimal and common support devices (i.e., computer, tablets, phone...etc.).

3. Economic

An immediate impact of the system would be improving the level of education of students from underprivileged environments. The final product will be given to the NGO Ed4.0 to be used in their research and work on improving education throughout Morocco, particularly in Moroccan villages. The NGO typically provides the hardware devices whenever it runs its tests and/or model classes. As the software will not generate any profit, no external economic factors will affect its development and/or delivery.

4. Environmental

There are no direct links between the impact of this system and the environment. An implicit connection could potentially be the impact of digitalized learning on sparing paper and other material involved in traditional education. As the ITS will not be a substitute to formal education in general, this effect is expected to be minimal. It is also worth mentioning that digital devices themselves incur a cost on the environment because of their reliance on electricity and any additional manufacturing resources, engendering disposal of potentially toxic batteries, that go into their fabrication, and may not abide by the general environmental regulations on the field.

5. Political

The political implications of the system concern the effect of organizations on the project. Two organizations that are relevant to this project are the Ministry of Education in Morocco and the NGO Ed4.0. To the best of my knowledge, the Ministry of Education does not have any laws on the use of educational software in general. In general, the authorities seem to be supportive of initiatives introducing Information and Communication Technology (ICT) to Moroccan schools, so we suspect their support of their project if we ever come to need it. Any large-scale work in this sense will be done in consultation with the concerned authorities from the Ministry. Regarding the NGO, their use of the final product shall be subjected to a contract detailing property rights and conditions of use.

6. Legal

The legal factors relevant to this capstone involve the use of data collected from learners and any international or national laws regulating the practice. As of now, the data involved in the system is generic enough not to raise privacy issues or other legal concerns in general. Overall, educational systems tend not to collect data that has the potential of harming the users if it were to be released publicly. The scope of this capstone does not cover any testing with real-students, so these concerns relate more to future work. We will further consult relevant specific laws before running experiments.

7. Ethical

Ethical concerns revolve around much of the same factors as their legal counterparts: privacy concerns of data collection. Any work in this regard will abide with the moral and legal codes in practice within the context of use of the software. Another ethical issue concerns the likely disparity in access to educational technology among pupils of different socio-economic

backgrounds. Luckily, with the support of initiatives like Ed4.0, and the continuous decrease in the price of technological devices in general, the gap in accessing technology will shrink rapidly over time. In the meanwhile, the platform will be developed with this accessibility concern in mind (i.e., by providing support to the most common and cheapest technological devices). The use of free-access/open-source tools goes towards fulfilling this aim.

CONCLUSION AND FUTURE WORK

This project explored the creation process of an ITS from pedagogical domain analysis to the final implementation of the system. In addition to its interdisciplinarity, this process puts to use a great number of academic skills, technical and literary alike. A review of the state-of-the-art research on the pedagogy of teaching early algebra was investigated, then applied to analyze the Moroccan 7th grade math curriculum as the main inspiration for the content of the ITS. The final lesson plan proposed for this system was analogous to its source of inspiration with small improvements in introductory approaches. An overview of the field of ITS development and its recent advances, along with a description of the technology used for the implementation of the final system, preluded a presentation of the final product with its three modules and main exercise types. At this point, the final system is in the form of an online tutor offering support for students studying in the Moroccan system, either as an online exercise platform or a stand-alone educational environment for personal learning.

Although the current state of the system is overall satisfying in terms of its development, a lot of work is yet to be done on testing and subsequent improvements of the content and implementation of the tutor. The project will be released as open-source for the community to get involved in finishing this work in case I do not manage to finish it this summer. A first-round of testing should aim to check the behavior of the current production rules and how well they scale up to problems of similar nature. The second ‘official’ testing should be done with a real-audience in order to get an assessment of the effects of this form of tutoring on Moroccan middle school students. Pre- and post-performance data will be recorded and reported through adequate statistical methods. The input of educators on the system will also be collected to inform any adjustments or improvements required to make the system a better fit for its audience. Once a stable and effective version of the software is reached, development shall move towards expanding the curriculum to offer a wider range of practice problems and hence more room for adaptive sequencing. Finally, data about the performance of the current system and all its subsequent versions will be kept available through the platform DataShop for research and assessment purposes.

REFERENCES

- [1] M. C. Desmarais and R. S. J. d. Baker, "A review of recent advances in learner and skill modeling in intelligent learning environments," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 9–38, Apr. 2012.
- [2] E. Filloy and T. Rojano, "Solving Equations: The Transition from Arithmetic to Algebra," *For the Learning of Mathematics*, vol. 9, no. 2, pp. 19–25, Jun. 1989.
- [3] K. R. Koedinger and B. A. MacLaren, "Developing a Pedagogical Domain Theory of Early Algebra Problem Solving," Carnegie Mellon University, Pittsburgh, tech., 2002.
- [4] B. A. Jimenez, G. R. Courtade, and D. M. Browder, "Teaching an Algebraic Equation to High School Students with Moderate Developmental Disabilities," *Education and Training in Developmental Disabilities*, vol. 43, no. 2, pp. 266–274, Jun. 2008.
- [5] S. J. PAYNE and H. R. Squibb, "Algebra Mal-Rules and Cognitive Accounts of Error," *Cognitive Science*, vol. 14, pp. 445–481, 1990.
- [6] 2018. المفيد في الرياضيات ,السنة الأولى من التعليم الثانوي الإعدادي ,إحمد ابو شادي
- [7] 2005. الجيد في الرياضيات ,السنة السادسة من التعليم الأساسي الابتدائي صالح بعيز
- [8] البرامج والتوجيهات التربوية الخاصة بمادة الرياضيات بسلك التعليم الثانوي الإعدادي
- [9] V. J. Shute and Joseph Psotka, "Intelligent Tutoring Systems: Past, Present, and Future," tech., 1994.
- [10] J. R. Anderson, C. F. Boyle, A. T. Corbett, and M. W. Lewis, "Cognitive Modeling and Intelligent Tutoring," *Artificial Intelligence*, vol. 42, pp. 7–49, 1990.
- [11] G. I. Webb, M. J. Pazzani, and D. Billsus, "Machine Learning for User Modeling," *User Modeling and User-Adapted Interaction*, vol. 11, pp. 19–29, 2001.

- [12] M. Ochs and C. Frasson, “Emotionally Intelligent Tutoring Systems (EITS),” tech.
- [13] Cmuctat, “Building an HTML Student Interface,” *Github - CTAT Wiki*, 10-Jan-2019. [Online]. Available: <https://github.com/CMUCTAT/CTAT/wiki/Building-an-HTML-student-interface>. [Accessed: 22-Apr-2019].
- [14] Cmuctat, “Quick Start,” *Git Hub - CTAT Wiki*, 01-Nov-2019. [Online]. Available: <https://github.com/CMUCTAT/CTAT/wiki/Quick-Start>. [Accessed: 22-Apr-2019].
- [15] Cmuctat, “Example-Tracing Tutors,” *Git Hub - CTAT Wiki*, 10-Jan-2019. [Online]. Available: <https://github.com/CMUCTAT/CTAT/wiki/Creating-an-example-tracing-tutor>. [Accessed: 22-Apr-2019].
- [16] Cmuctat, “Creating Cognitive Tutor,” *Git Hub - CTAT Wiki*. [Online]. Available: <https://github.com/CMUCTAT/CTAT/wiki/Creating-a-Cognitive-Tutor>. [Accessed: 22-Apr-2019].
- [17] Cmuctat, “Behavior Graphs,” *Git Hub - CTAT Wiki*, 25-Aug-2017. [Online]. Available: <https://github.com/CMUCTAT/CTAT/wiki/CTAT-Behavior-Graphs>. [Accessed: 22-Apr-2019].
- [18] R. Nkambou, J. Bourdeau, and Riichiro Mizoguchi, “Introduction: What Are Intelligent Tutoring Systems, and Why This Book?,” *Studies in Computational Intelligence*, Jan. 2010.

APPENDIX A: PROFESORS' INTERVIEW SUPPORT SHEET (ENGLISH VERSION)

1. Definitions

- **Algebra:** refers to the field of mathematics dealing with equation solving. In the context of this research, algebra in mathematical education is defined as the step when students learn to **operate on entities other than numbers** to solve equations or problems.
- **Pre-Algebra:** generic term designating mathematical content meant to prepare students for developing algebraic skills before the formal teaching of Algebra. Such content includes introducing **informal strategies to solve arithmetic problems** such as undoing operations, guess-and-test...etc.
- **ITS:** stands for Intelligent Tutoring System and refers to a computer program providing **immediate and adaptive instructions and feedback to learners**, usually without the intervention of a human teacher.
 - Questions/Comments:

2. General Approaches for Teaching Algebra

2.1. Algebra ILO:

- General ILO:
 - Acquiring the language of algebra
 - Modeling word problems into mathematical equations and solving them correctly
 - Other ILO?
- Specific ILO: (see Table A1)
- General views on teaching Algebra:

There exist two general views on teaching Algebra: The Syntactic and the Semantic Approach.

- **Syntactic:** teaching algebra rules to be applied to equations in an abstract context.
- **Semantic:** endowing equations with meaning to deduce algebraic rules using contextual cues.

Table A1: Algebra ILO for Interview Support Sheet

	Example	Comment
Equality	<ul style="list-style-type: none"> - Arithmetical definition of equality: left-side represents actions, right-side consequences => unwinding/undoing operations leads to a result. - Transposition of the two-sides of the equation is not evident, as suggest studies on pre-symbolic algebra as well as work with students in introductory algebra levels. 	
Operating on the unknown	<ul style="list-style-type: none"> - When at least two instances of the unknown are present in a first-degree equation, solving the equation becomes more challenging for students. It seems that performing arithmetic operations on entities other than numbers is a concept to be introduced separately. (e.g $4x + 6 = 2x$, equation requiring more than inverting the operations of the coefficients). 	
Concept 1:		
Concept 2:		

- What is your experience with both?
- Which do you typically start with?
- Which is more successful with students? How do different students react to each method?
- Additional input:

2.2. Teaching Strategies:

- What are some of your teaching strategies for introducing key algebra concepts?

- Example: geometric modeling of quantities, balance example, replacing the quantities in an equation with small numbers to deduce the arithmetic operations to perform, etc.
- Take notes on Table A2.

Table A2: Notes on Proposed Algebra Strategies

Name of Strategy	Concept Introduced	Description & Example
Strategy 1:		
Strategy 2:		
Strategy 3:		
Strategy 4:		

3. Prerequisites for studying algebra

- Arithmetic Skills (see Table A3):

Table A3: Arithmetic Prerequisites for Teaching Algebra

Prerequisite	Description & Example
Arithmetic operations (+, -, *, /) on integer values	<p>Description: 4 basic arithmetic operations on 1 or more integer numbers, leading to integer or decimal results (in the case of division)</p> <p>Example:</p> <p>$5+5$, $6*9$, $22 - 59 + 69$</p>
Modeling and solving arithmetic problems (result-unknown problems) [put as separate skills]	<p>Description: problems in which a student is operating on numbers to find a missing value. No formal understanding of equation modeling is required to solve the problem.</p> <p>Example:</p> <p>Salma wants to buy 4 fruits, each one costs 2 dhs, how much did she pay in total?</p> <p>There are 8 blue crayons and 7 red crayons in a box. Miss Scott adds some more crayons to the box. Now there are 19 crayons in the box.</p>
Prerequisite 1:	
Prerequisite 2:	

4. Common students' difficulties and errors

- Two major types of student errors: arithmetic and conceptual errors.
 - Arithmetic errors include calculation errors ($1+2 = 4$), operation precedence errors ($2*3+1 = 6$), and any other errors which can be found in earlier arithmetic lessons.
 - Conceptual errors are specific to Algebra concepts. These include performing an operation on a single side of an equation, combining non-like terms ($2x + 4 = 6x$).
- From your experience, what errors do students make the most when working with Algebra for the first time?

6. What do you wish to have in an ITS as an educational support software?

- Do you think an ITS can be used to introduce algebra?
- Do you think the ITS alone can teach algebra? If yes, how?
- Which teaching strategies do you recommend using in the software?
- What kind of observations should be taken into consideration for making the software adaptive?

APPENDIX B: ITS STUDENT INTERFACES

Problem 1:

This first problem is meant to introduce the mathematical translation of word equations as closely as possible. This exercise will be prefaced by a guided solution of a problem of the same type, meant to explain the solution steps (algebraic manipulations) expected of the learner in this context.

الدرس الثاني

حل المعادلة الآتية: للحصول على 35، طرحنا 2 من عدد ثم أضفنا إليه 5. ما هو هذا العدد؟

x - 2 + 5 = 35

+ +

+ + + +

+

العدد هو :

تحقق من النتيجة: تحقق من صحة النتيجة المحصل عليها.

+ + +

Figure 19: Interface for Module 1 Problem 1

Problem 2:

This level of difficulty aims to test the student's familiarity with basic arithmetic rules such as associativity, commutativity and precedence of arithmetic operations. To this aim, the order of the terms of the LHS of the equation is mixed-up compared to the previous level. This would become more relevant for problems including multiplication or division operations.

الدرس الثاني

حل المعادلة الآتية: للحصول على 35، طرحنا 2 من عدد ثم أضفنا إليه 5. ما هو هذا العدد؟

$$5 - 2 + x = 35$$

Figure 20: Interface for Module 1 Problem 2

Problem 3:

At this level, the student should be exposed to the symmetry of algebraic equality, by working through an equation of reversed left- and right-hand sides. The problem in this case may or may not include an intermediate step in which the student can revert the equation to a form that makes more sense to him/her. Eventually, the students should be able to solve equations regardless of their initial direction/order of sides.

الدرس الثاني

حل المعادلة الآتية: للحصول على 35، طرحنا 2 من عدد ثم أضفنا إليه 5. ما هو هذا العدد؟

$$35 = x - 2 + 5$$

Figure 21: Interface for Module 1 Problem 3

Problem 4:

For this last problem type, the student would be faced with an interface partially prefilled at different levels, and asked to work on the equation with its initial ‘constraints’. This is meant to test the student’s mastery of the previously introduced concepts and their comfort with all algebraic manipulations regardless of the order of the terms and sides.

الدرس الثاني

حل المعادلة الآتية: للحصول على 35، طرحنا 2 من عدد ثم أضفنا إليه 5. ما هو هذا العدد؟

<input type="text"/>	<input type="button" value="+"/>	<input type="text"/>	<input type="button" value="x"/>	<input type="button" value="+"/>	<input type="text"/>	<input type="button" value="+"/>	<input type="text"/>
<input type="text"/>	<input type="button" value="+"/>	<input type="text"/>	<input type="button" value="+"/>	<input type="text"/>	<input type="button" value="+"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="button" value="+"/>	<input type="text"/>	<input type="button" value="x"/>	<input type="button" value="+"/>	<input type="text"/>	<input type="button" value="="/>	<input type="text"/>
<input type="text"/>	<input type="button" value="+"/>	<input type="text"/>	<input type="button" value="x"/>	<input type="button" value="="/>	<input type="text"/>	<input type="button" value="+"/>	<input type="text"/>

Figure 22: Interface for Module 1 Problem 4