UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project Final Paper 2023

Title: An Ontology-Based Adaptive Learning System to Identify Learning Gaps

Author: Aakief Hassiem

Project Abbreviation: GALMAT

Supervisor(s): Toky Raboanary

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	10
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	15
System Development and Implementation	*0*	*20*	10
Results, Findings and Conclusions	*10*	*20*	15
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	

An Ontology-Based Adaptive Learning System to Identify Learning Gaps

Aakief Hassiem
University of Cape Town
Cape Town, South Africa
HSSMUH018@myuct.ac.za

ABSTRACT

With the rapid expansion of technology, conventional teaching approaches have grown outdated, prompting educational institutions to embrace e-learning. Infusing classrooms with technology yields several advantages, including heightened learner engagement, cultivating digital literacy, and facilitating interactive learning encounters. However, e-learning grapples with challenges such as accommodating diverse learning styles and addressing areas of vulnerability in course materials. Active research endeavours are ongoing to develop systems which provide each learner with unique educational materials based on their academic ability.

This research paper introduces an ontology-based adaptive learning system to assess learners' proficiency levels of concepts within a specific knowledge domain. The primary aim is to uncover potential knowledge gaps a learner might possess. While the adaptive system demonstrated the ability to gauge learners' abilities accurately, its performance exhibited inconsistency, warranting necessary improvements. The contributions of the adaptive system provide unique insights, particularly in conceptualising test evaluation using item response theory.

CCS CONCEPTS

• **Applied computing** → Computer in other domains; Education; • **Theory of computation** → Theory and algorithms for application domains.

KEYWORDS

E-learning, Ontology, Item Response Theory, Adaptive learning

1 INTRODUCTION

Recent technological advancements have enabled the emergence of new learning delivery methods [17]. The integration of computers into educational settings, often referred to as electronic learning or e-learning, has reshaped how education is approached. E-learning, defined as "learning conducted via electronic media, typically on the Internet" [20], has revolutionised education by making online learning materials highly accessible. This shift has far-reaching implications, not only affecting learners but also benefiting educators through improved communication and efficient coursework delivery [15]. Universities have embraced e-learning through portals and online platforms such as Microsoft Teams, Zoom, and Google Meet, enhancing interactions between students and instructors. The increasing prevalence of e-learning in universities highlights the need for more adaptable educational delivery methods to keep pace with transformative changes [8].

E-learning platforms often rely on a "one-size fits all" approach, which can present limitations in terms of customisation, flexibility, and compatibility and is widely considered to be a significant weakness of this type of education system [2]. Despite providing identical course content to all students, this approach neglects the individualised learning styles and abilities of each student [9]. The notion of learning styles suggests that people have differing methods of learning and retaining knowledge, and it is imperative to recognise a student's strengths and weaknesses in this regard [16]. Moreover, the rate at which students learn and comprehend new material can vary, which can put some students at a disadvantage if e-learning platforms do not account for these discrepancies. Additionally, students may encounter challenges in specific subtopics, leading to areas of weakness in their knowledge [5].

The current focus of scientific research in e-learning is on developing learning platforms that meet the diverse expectations, motivations, learning styles, habits, and needs of learners [10]. To achieve this, adaptive e-learning systems have emerged that provide personalised learning materials based on individual student characteristics. It's worth noting that traditional e-learning systems that offer the same materials to all students do not offer personalised learning experiences [19]. Given that each learner has unique strengths and weaknesses in course content, an adaptive e-learning system should identify and address these weaknesses by selecting appropriate learning materials to create a tailored learning experience.

Our objective is to develop an adaptive e-learning system that can accurately assess the proficiency of students across various concepts, leveraging an ontology as its foundation of knowledge. This system will offer personalised suggestions to individuals who may require additional guidance in certain areas.

2 RELATED WORK

2.1 Computerised Adaptive Testing

Computerised Adaptive Testing (CAT) is a computer-based assessment method designed to simplify the testing process for examinees [3]. It is particularly helpful for longer tests that would otherwise require a substantial amount of time and effort to prepare for [22]. CAT works by estimating an examinee's trait level (θ) and selecting relevant test items from a pool of items [14]. In most CATs, item selection and proficiency evaluation are based on item response theory (IRT) [21]. For further details on IRT, please refer to Section 3.1.

The CAT assessment comprises two primary stages. Initially, it carefully chooses the starting items based on the individual's

, September 2023, CPT, South Africa
2023. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

estimated abilities. Following this, it evaluates the examinee's responses by utilising a measurement technique such as IRT. Based on the score obtained, the estimated ability of the examinee is adjusted accordingly. This iterative process is continued until a predetermined stopping criteria is met, which is typically an optimal level of measurement accuracy or a set number of items. During the entire process, the CAT algorithm generates a conclusive evaluation of the individual's ability, providing a comprehensive and accurate assessment [22].

Computer Adaptive Testing (CAT) shows promise in streamlining the testing process, but there are hurdles that CAT developers and managers must overcome, as outlined by Wise in their article [21]. We can group these obstacles into four main categories: The first is "Item Pool Development and Maintenance", which involves managing the item pool and selecting the appropriate IRT model. The second category is "Administering and Scoring the CAT", which covers the steps required to calculate a test-taker's ability level and score their test. The third category is "Protecting the Integrity of the CAT Item Pool", which emphasises the importance of security measures to safeguard the item pool and preserve the CAT's integrity. Finally, the fourth category is "Examinee Issues in CAT", which addresses the challenges that both managers and test-takers may encounter when using CAT. According to Wise, this last category is the most significant concern when dealing with CAT.

As per the research conducted by Latu and Chapman [11], CAT can be highly beneficial when fully utilised for tests that require instant scoring, customised content, and enhanced efficiency. However, using it for lengthy essay-based tests can result in a significant increase in administrative costs. Nevertheless, ongoing efforts are being made to reduce the cost of utilising CAT in real-world scenarios.

2.2 Personalised e-learning system

In a research study, Boyinbode, O. [1] created a web-based application that implements an ontology to tailor e-learning experiences. The objective of this approach is to produce relevant learning materials for each student based on their learning style, background knowledge, preferences, and personal profile. The system leverages Web Ontology Language (OWL), a semantic web language that captures knowledge about entities and their relationships. The OWL file is produced by the Protégé tool, which extracts classes or concepts from a domain ontology. The ontology-driven adaptive system [1] consists of multiple components, with the Personalised Adaptive Engine being the most critical.

The Personalised Adaptive Engine is responsible for creating customised learning materials for learners based on their unique learning models. It achieves this by combining instructional items to form organised content while gathering information about the learner and learning objects through intermediaries. The system continuously evaluates the learner's knowledge and abilities, utilising IRT to assess performance. This model-based approach selects the best learning items for the learner by analysing the relationship between their abilities and responses to the items.

To evaluate the effectiveness of the system, two test methods were conducted. The first method assessed individual learners' performance and tracked their learning progress over time. The second method utilised a General Study Course (GSC) as the learning material to test the personalised adaptive e-learning system. The results showed that the personalised adaptive system outperformed the conventional system, delivering tailored content to each learner.

Despite its success, the system's content is limited to what is available in the content model and cannot generate unique content. This limitation prevents it from filling all knowledge gaps. Therefore, the ability to automatically generate new questions related to a specific subtopic in the domain would enhance the learner's understanding of the material, especially for more knowledgeable students. It would also challenge them to their limits and prevent them from becoming bored with the system.

3 BACKGROUND

3.1 Item Response Theory

Item Response Theory (IRT), also referred to as latent response theory, is a sophisticated mathematical concept that sheds light on how individuals perform on tests [12]. By taking into account personalised and item-specific variables, IRT can determine the likelihood of a given individual providing the correct response to a given item. This information can then be used to identify the best learning items for the individual. In the case of Item Response Models (IRM), the focus is on the individual's traits, such as their ability level. The primary statistical tasks revolve around estimating model parameters and assessing the model's fit to the item responses. IRM can transform qualitative data (like test results) into quantitative data through the use of ability parameters and item parameters, both of which describe the properties of the items that impact the examinee [6].

When utilising IRT, it is crucial to take into account three significant assumptions. Firstly, IRT is unidimensional. It is designed to measure a singular trait or ability of a participant and cannot be used to evaluate multiple abilities at once. For instance, attempting to measure a student's math ability and their language ability using IRT simultaneously would not be appropriate. Secondly, there is local independence. It is vital to ensure that the participant's response to one item does not affect their response to another item in the test. This enables the model to analyse each item independently. Lastly, there is parameter invariance. This means that the difficulty level and the degree to which items distinguish between participants should remain consistent, regardless of whether they are of high or low ability. These assumptions are vital to keep in mind while using IRT to attain accurate results [13].

It is possible to use IRT with various types of data such as polytomous, continuous and dichotomous data. The system presented in this research paper would use dichotomous data as input. Dichotomous data refers to data that permits only two possible responses. Some common examples of dichotomous data include true-or-false questions and multiple-choice questions with only one correct answer. To handle dichotomous data, there are three IRT models available: the one-parameter, two-parameter, and three-parameter models. These models utilise a logistic function, also known as an "Item Response Curve" (as displayed in Figure 1), to estimate the probability of an individual providing a correct response [12].

As demonstrated in Figure 1, there is a clear correlation between ability level and correct response rate. Notably, at an ability level of

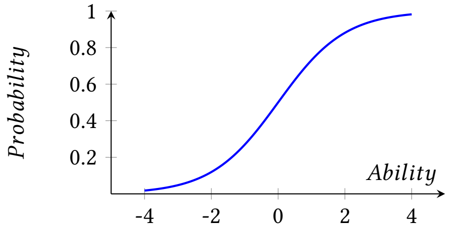

Figure 1: Item Response Curve for Dichotomous Data. This figure illustrates a typical Item Response Curve (IRC), a core concept in IRT that is used to analyse and describe the link between an individual's ability and performance on test items. The curve represents dichotomous data, in which each item has only two possible replies, and it illustrates how different test items perform in relation to individual talents.

0, the likelihood of answering correctly is 50%. The curve's shape is determined by a unique set of parameters for each IRT model.

3.1.1 1 parameter model.

$$P_{ij}(\theta_j, b_i) = \frac{e^{(\theta_j - b_i)}}{1 + e^{(\theta_j - b_i)}} \tag{1}$$

Where,

θ_i = ability

b_i = difficulty parameter

One of the simplest models for IRT is the Rasch model, also known as the one-parameter model. This model proposes that a test taker's ability to answer an item correctly is influenced solely by the difficulty of the item. The item difficulty value determines the position of the curve on the x-axis. A lower item difficulty value shifts the curve to the left, indicating that a lower ability level is required to answer the item correctly. Conversely, a higher item difficulty value shifts the curve to the right, indicating that a higher ability level is necessary to answer the question correctly. The difficulty parameter is rated on a scale of +2 to -2, where +2 refers to high-difficulty items and -2 refers to low-difficulty items.

3.1.2 2 parameter model.

$$P_{ij}(\theta_j, b_i, a_i) = \frac{e^{a_i(\theta_j - b_i)}}{1 + e^{a_i(\theta_j - b_i)}} \tag{2}$$

Where,

a_i = Discrimination parameter

The two-parameter model incorporates a discrimination parameter alongside the difficulty parameter, employing two item characteristics to enhance accuracy. The discrimination parameter gauges an item's ability to distinguish between individuals with differing levels of ability, resulting in more precise evaluations. A higher value for this parameter corresponds to a steeper slope, while a lower value corresponds to a gentler slope. Although two-parameter models offer greater precision, they usually require more data than a one-parameter model.

3.1.3 3 parameter model.

$$P_{ij}(\theta_j, b_i, a_i, c_i) = c_i + (1 - c_i)\frac{e^{a_i(\theta_j - b_i)}}{1 + e^{a_i(\theta_j - b_i)}} \tag{3}$$

Where,

c_i = Guessing parameter

The most complex IRT model is the three-parameter model, which includes a guessing parameter to account for any guessing on an item. This parameter shifts the item response curve along the y-axis. While the three-parameter model has the potential to yield a more accurate test score, it requires a larger dataset compared to the two-parameter model. However, the improvement over the two-parameter model is only marginal.

IRT has many practical applications in scoring and test-making. Once IRT parameters are calculated, they can be used to estimate the test-takers ability, traits, and test item difficulty. IRT assists with item selection, bias detection, and adaptive testing. It can also identify test items specifically for a target population and check for biases by associating them with the performance of various subgroups on the same items. Furthermore, IRT is useful in producing scaled scores that take into account the difficulty of a question, unlike traditional test scoring methods [12].

4 ONTOLOGY-BASED ADAPTIVE LEARNING SYSTEM

4.1 Approached Architecture

The adaptive system consists of several components, as shown in the Appendix Section, Figure 9. Each subsequent component in the adaptive system takes the output of the previous component as its input to deliver the final output. A high-level algorithm for the adaptive system is shown in Listing 2.

Listing 2 presents pseudo-code outlining the algorithm employed by the adaptive system. This algorithm encompasses four principal methods: '*Run assessment unit*', '*Run IRT unit*', '*Update learner ability dictionary*', and '*Obtain triples*'. Each of these methods corresponds to and offers a high-level overview of the functions of a critical component in the system's architecture, as depicted in Figure **??**. Specifically, these components relate to '*Assessment unit*', '*IRT unit*', '*Learner ability dictionary*', and '*Triples*', respectively. As illustrated above, there is a clear flow of data between these components, with the output of one method serving as input for another. In detail, *Line 6* returns the *testResults* object from the '*Run assessment unit*' method, which is subsequently utilised as a parameter in *Line 8* for the '*Run IRT unit*' method. The sequential nature of the algorithm is evident in *Lines 17-18*, where '*Run IRT unit*' follows '*Run assessment unit*', and this pattern continues in *Lines 18-19* between '*Run IRT unit*' and '*Update learner ability dictionary*'. Notably, '*Obtain triples*' can only be executed after the learner ability dictionary has been updated, maintaining the sequential pattern. The *while* loop in *Lines 16-21* signifies that as long as the adaptive system receives valid input, these four methods will be executed. This is necessary because, in adaptive systems like CAT (as discussed in Section 2.1), multiple tests must be conducted to obtain an accurate assessment of the learner's abilities.

```
Input : Vector in the form: [concept, answer, memo, difficulty]
Output : Learner knowledge model which contains triples
procedure Run Assessment Unit(Input vector)
    Assess test results 1: Correct; 0:Incorrect
    Calculate $difficultyFactor$
    Return $testResults$ vector
end procedure
procedure Run IRT Unit($testResults$)
    Perform IRT calculations for each concept
    Return $LearnerAbilities$
end procedure
procedure Update learner ability dictionary($LearnerAbilities$)
    Update learner abilities for each concept
    Update neighbouring concepts learner abilities
end procedure
while $input$ do
    Run assessment unit
    Run IRT unit
    Update learner ability dictionary
    Obtain $triples$
end while
```

Figure 2: Pseudo-code for the Adaptive System: This figure provides an overview of the adaptive system's functionality through pseudo-code. It illustrates the four key methods and their roles in the system's execution. Additionally, it outlines the input and output processes, shedding light on the system's inner workings.

4.2 Assessment Unit

The assessment unit is responsible for processing the test results for the learner. The test results of a learner are a nested vector provided by an Automatic Question Generator (AQG). The format of the input is,

$$\begin{array}{llll}
[\text{concept}_1, & \text{answer}_1, & \text{memo}_1, & \text{difficulty}_1] \\
[\text{concept}_1, & \text{answer}_2, & \text{memo}_2, & \text{difficulty}_2] \\
\vdots & \vdots & \vdots & \vdots \\
[\text{concept}_n, & \text{answer}_m, & \text{memo}_m, & \text{difficulty}_m]
\end{array}$$

Where *concept* is the question's subject, *answer* is the learner's response, *memo* is the correct response, and $difficulty$ is the question's level of difficulty, with 1 being a challenging question and 0 denoting an easy one. The n and m variables represent the concepts and questions, respectively. Because it is extremely unlikely that only one question would be asked for each concept, it is important to notice that the likelihood that n equals m is close to zero.

Processing the input consists of three operations. The first operation is checking whether the student answered the question correctly. For a correct answer, a 1 is assigned, and for an incorrect answer, a 0 is assigned to the question. The second operation is grouping the responses per concept and storing them in a vector. For example, if we had the following vectors for concept x,

$$\begin{array}{llll}
[\text{concept}_x, & \text{True}, & \text{False}, & 1] \\
[\text{concept}_x, & \text{True}, & \text{True}, & 0]
\end{array}$$

The resulting grouped vector for concept x would be,

$$[\text{concept}_x, \quad 0, \quad 1]$$

This type of vector is also known as a response vector (Y_1, Y_2,Y_3,...,Y_n) where $Y_i \in \{0, 1\}$.

The third operation is calculating a $difficultyfactor$. The difficulty factor takes into account the number of difficult questions a learner has answered correctly. The motive behind a difficult factor is to give students who answer difficult questions a higher score for that question, which would hence boost their learning ability. For example, without a difficulty factor, if a learner x were to answer two easy questions correctly and a learner y were to answer two difficult questions correctly, the system would calculate their learner abilities to be equal (for those two questions), when in reality learner y's learner ability should be higher. The difficulty factor uses the difficulty value in the input vectors, as described above. The equation for calculating the difficulty factor is,

$$\begin{aligned}
\text{Let } m &= \text{Number of correct difficult questions answered correctly} \\
&\quad \text{for concept}_x \\
\text{Let } n &= \text{Total number of difficult questions received for} \\
&\quad \text{concept}_x
\end{aligned}$$

$$difficultyfactor = \frac{m}{n} * 0.1 \tag{4}$$

Once the difficulty value is calculated for $concept_x$ it is added to its response vector. Leaving a resulting vector for $concept_x$,

$$[\text{concept}_x, \quad \text{difficulty factor}, \quad 0, \quad 1]$$

Each test concept would be assigned its own response vector. These response vectors are subsequently passed to the Item Response unit, which computes the learner's abilities for each concept.

$$\begin{array}{lrrr}
[\text{BarnyCakes}, & 0.05, & 0, & 1] \\
[\text{Bakso}, & 0.1, & 1, & 1] \\
[\text{Bionico}, & 0.05, & 1, & 0]
\end{array}$$

Figure 3: Example of assessment unit output. The concepts used in the test are *BarnyCakes*, *Bakso*, and *Bionico*. 1 out of 2 questions were answered correctly for *BarnyCakes* and *Bionico*, while both questions were answered correctly for *Bakso*. We can assume that both questions are difficult.

4.3 Item Response Theory (IRT) unit

From research and literature, as discussed in Section 2, there has been no attempt to use IRT to calculate the learner abilities for each concept in a test; rather, IRT has been used to calculate the learner abilities for the whole test. This distinct difference is what sets aside the adaptive system presented in this research paper from existing adaptive systems.

The IRT unit performs IRT calculations on each concept in the test. The IRT unit takes the response vectors for each concept shown in Section 4.2 and calculates their respective learner abilities. The IRT calculations were done using a Python module, "Pyirt". Pyirt is a useful module when performing IRT calculations as it has the ability to estimate various model parameters required for obtaining the value of the learner's ability, θ.

Once the learner's ability, θ, is estimated, the difficulty factor is added to it to account for the difficult questions in the test, as described in Section 4.2.

$$LearnerAbility_{final} = LearnerAbility_{IRT} + difficultyFactor \quad (5)$$

The final learner abilities for each concept are stored in a vector.

4.4 Learner ability bank

The learner ability bank calculates the final learner abilities for each of the concepts using the formula in Equation 5. Once the final learner abilities have been obtained for each concept the resulting learner ability vector would be,

$$\begin{array}{ll} [\text{concept}_1, & FinalLearnerAbility_1] \\ [\text{concept}_2, & FinalLearnerAbility_2] \\ \vdots & \vdots \\ [\text{concept}_n, & FinalLearnerAbility_n] \end{array}$$

The learner ability vector is used to update the learner ability dictionary.

4.5 Ontology handler

The ontology handler is responsible for performing any operations that involve the ontology. The main operations that the ontology handler performs are retrieving linking nodes and properties and searching for nodes.

The ontology handler also serves as a way for the learner ability dictionary to retrieve and initialise the subjects of the knowledge domain. Furthermore, the ontology handler is also used to update the learner ability dictionary, as described in Section 4.6. When updating the learner ability dictionary, the ontology handler updates the learner abilities for each concept in the vector shown in Section 4.4 by using the corresponding learner ability. The formula for updating the learner ability is,

$$UpdatedAbility = 0.25 * OldAbility + 0.75 * NewAbility \quad (6)$$

which is a weighted sum where more weight is given to the new ability due to it being the latest learning ability and more relevant to the current knowledge of the student about the concept.

The last major function that the ontology handler performs is generating triples for each of the concepts in the test and writing them to a CSV file, the output of the adaptive system. See Section 4.7 for the discussion on triples.

Overall, the ontology handler is a key component that allows all the other components in the adaptive system to communicate and interact with each other.

4.6 Learner ability dictionary

The learner ability dictionary stores each subject in the knowledge domain with its corresponding learner ability. In this research paper, a subject is defined as a non-leaf in an ontology, as shown in Figure 4. Hence, only subjects could be concepts when asking any question.

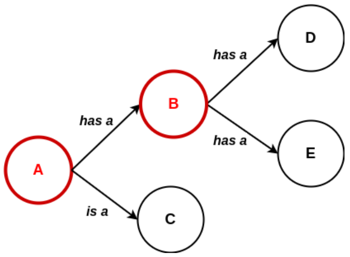

Figure 4: Directed Graph highlighting subjects A and B. A and B are subjects due to them being non-leaf nodes. Nodes D, E and C are not subjects due to them being leaf nodes (nodes with no outgoing arrows)

When the adaptive system is first run, all the subjects' learner abilities are initialised to 0. When the IRT unit calculates the learner's ability, it would fall in the range of -4 and 4. Setting the initial value of the learner's ability to 0 implies that when the learner takes the first test, we assume their knowledge about each concept is average. This would give us an initial impression of the learners' degree of knowledge about a given concept when asked about it for the first time. This is important as it dictates what type of questions the learner will receive when asked about the concept again.

In addition to updating the subject's learner abilities, the learner ability ontology also updates any neighbouring subjects learner abilities. In Figure 4, we see that a neighbouring subject of A is B. Neighbouring subjects are also called correlating concepts and are discussed further in Section 4.7.

4.7 Triples

Triples are used to represent knowledge. It describes relationships between different elements in a knowledge graph, such as an ontology. A triple consists of three objects: a subject, a predicate, and an object. The subject is the concept that the triple explains. A predicate, otherwise known as a property, is the relationship that connects the subject and object. Lastly, the object is the target entity associated with the subject. In the WebNLG dataset, triples are represented as

$$subject|predicate|object$$

and can be used to form ontologies as described in Section 4.8.1.

The association between the subject and object in a triple could also be viewed as correlating concepts, where the subject influences the object. From Figure 4, we can see that taking A as the subject, the two correlating concepts would be B and C. This implies that changes to the value of A influence the values of B and C.

When analysing a student's understanding of a particular concept, the logic of correlating concepts is important. If a learner struggles with questions relating to $concept_X$ and $concept_X$ influences $concept_Y$, by extension, we can conclude that the learner would to some degree struggle with $concept_Y$. This can be seen in Figure 5, $concept_X$ is directly related to $concept_Y$, hence the learning ability of $concept_X$ would affect the learning ability of $concept_Y$. The formula for calculating the updated learner ability for the correlating object is:

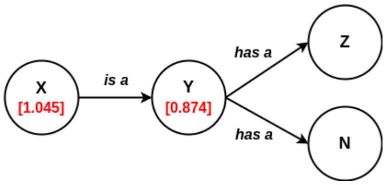

Figure 5: Graph depicting correlating concepts X and Y. Due to node X being connected to node Y, its value has a slight influence on the value of node Y. It's important to note that only nodes X and Y have values, as they are the only subjects in the graph.

$$\text{Let } m = \text{Old object ability}$$
$$\text{Let } n = \text{New subject ability}$$
$$UpdatedObjectAbility = 0.7 * m + 0.3 * n \tag{7}$$

The weighted sum places more emphasis on the object's learner ability, as although the subject's learner ability affects the object's learner ability, it should not overpower it.

From Figure 5 we can see that nodes Z and N contain no learner ability, and that is due to them being leaf nodes in the graph and therefore not being subjects/concepts.

4.8 Implementation

4.8.1 ***Knowledge domain.*** The knowledge domain was created using a WebNLG data set that contained triples of food. An ontology was constructed from the WebNLG data set using an open-source ontology editor, Protégé. WebProtégé is a Web-based ontology development environment that allows you to easily construct, upload, modify, and share ontologies for collaborative viewing and editing. WebProtégé has additional features such as a visualisation tool that allows users to view the ontology, which is particularly helpful when creating larger ontologies. Most importantly, WebProtégé supports the use of the OWL 2 web ontology language. This enables us to use Python modules that can easily manage ontology operations.

4.8.2 ***Managing ontologies in python.*** The W3C Web Ontology Language (OWL) is a tool of the Semantic Web created for describing intricate knowledge about objects, collections of objects, and connections between them. Owlready2 is a Python ontology-driven programming library with a primary focus on OWL ontologies for knowledge representation. It integrates with OWL, allowing developers to work with ontologies while using Python's language features. Owlready2 supports ontology reasoning, ontology creation, and import capabilities, supports SPARQL queries, and integrates with other Python libraries for data analysis and machine learning. Owlready2 is a useful tool for introducing semantic reasoning and knowledge representation into Python applications.

Owlready2 is used by the ontology handler to perform various functions relating to the knowledge domain, as described in Section 4.5.

4.8.3 ***Pyirt.*** Pyirt is a Python implementation of Item Response Theory. Pyirt only deals with a unidimensional theta, which means that all the items in the test are only measuring one latent trait, theta. There are two methods that can be used to estimate theta, θ. Bayesian estimation and Maximum Likelihood Estimation (MLE). By default, Pyirt uses MLE to calculate theta and all model parameters such as difficulty, discrimination, and the guessing parameter. To calculate the model parameters using MLE, Pyirt uses the Expected Maximisation (EM) algorithm.

The EM algorithm offers a method for calculating maximum likelihood and Bayes modal parameter estimates when dealing with scenarios involving incomplete data. The EM algorithm generates parameter estimates that optimise the likelihood of the observed data by calculating the likelihood of the entire dataset, which includes both the observed and missing data [7]. The EM algorithm consists of iterating over two steps, the E-step and the M-step. The E-step predicts a probability distribution across missing data completions given the present model. It originated because the probability distribution over completions is often not required to be explicitly created. Instead, it entails figuring out expected statistics that are sufficient for these completions. The M-step re-estimates the model parameters using the completions from the E-step. M-step refers to the idea that re-estimating the model can be thought of as maximising the expected log-likelihood of the data [4]. The E and M steps are repeated until the parameter estimations converge. The relative difference in observed data likelihood between successive iterations, or differences in parameter estimations over iterations, can be used to determine convergence [7].

By default, Pyirt uses a 2-parameter IRT model (Section 3, however, there is an option to include a guessing parameter. Since the parameters of an IRT model are continuous [7], bounds on the theta, difficulty, and discrimination parameters have to be set to avoid overfitting.

5 EVALUATION

The adaptive e-learning system was evaluated using three methods: qualitative analysis, quantitative measurement, and performance evaluation. This multifaceted approach gave a thorough understanding of the system's effectiveness, user experience, and operational efficiency. A qualitative study gathered user actions, emotions, and responses to system outcomes, identifying opportunities for improvement that quantitative data may not have revealed. The quantitative analysis gave objective information on system usage and efficacy. The performance evaluation focused on the system's operational efficiency, measuring speed and responsiveness to ensure a consistent user experience and identify any bottlenecks. The incorporation of these methods enabled informed judgements and improvements.

5.1 Accuracy of the System: Qualitative Analysis

The primary aim of this research study is to identify knowledge gaps within an individual's domain. Consequently, ensuring the adaptive system's accuracy in estimating individual abilities for creating customised educational content becomes paramount. To assess the precision of the adaptive system, a test was conducted involving six participants who answered two sets of true and false questions. To maintain authenticity, these questions were generated using the automatic question generator, a component integrated with the adaptive system (as detailed in Section 8.1). All questions for each concept were stored separately in their respective item banks.

Each test comprised 15 questions and covered three concepts from the food ontology: *Ajoblanco*, *Bacon Explosion*, and *Bhajji*. The evaluation unfolded through the following steps:

(1) **Question Generation and Storage:** Questions were created for each concept and stored for evaluation.
(2) **Standardised Test 1:** An initial standardised test was formulated.
(3) **Estimate Learner Abilities (Test 1):** Participants' abilities for each concept in the first test were estimated.
(4) **Question Selection and Test 2 Creation:** Based on participants' abilities in the first test, new questions were drawn from the item bank to construct a second test.
(5) **Estimate Learner Abilities (Test 2):** Participants' abilities for each concept in the second test were estimated.

After acquiring the estimated learner abilities for each concept from test two, participants were presented with a Likert scale, as shown in Table 1. They were then asked to rate the accuracy of the estimations for each ability. This rating was based on their perceived familiarity with the concepts and their self-assessment of their performance in answering the questions.

Concept	Accurate	Somewhat Accurate	Not Accurate
Ajoblanco		x	
Bacon Explosion		x	
Bhajji	x		

Table 1: The table gives an example of a Likert scale that has been completed by a participant after they have finished the two sets of test questions. They are asked to indicate using the Likert scale how accurately they thought the adaptive system performed when predicting their ability for *Ajoblanco*, *Bacon Explosion*, and *Bhajji*.

The results for the evaluation are shown in Table 2.

	Responses
Accurate	6
Somewhat Accurate	8
Not Accurate	4

Table 2: The table displays the results of the accuracy evaluation performed using qualitative analysis. The total responses for each accuracy level (accurate, somewhat accurate, or not accurate) were totaled across *Ajoblanco*, *Bacon Explosion*, and *Bhajji*.

5.2 Evaluating the Functionality and Adaptability of the System: Quantitative Analysis

To verify the functionality of the IRT unit, the aim is to validate its capacity to assign lower ability estimates to learners who struggle with a concept and higher ability estimates to those who perform better in another concept. This evaluation intends to compare the ability levels of concepts within a test against the corresponding test answers or response vectors. The underlying assumption is that the more correct responses a learner provides for a concept, the higher their ability level will be for that particular concept, and vice versa. Hence, this evaluation directly examines the outcomes of each individual concept.

As this assessment does not involve human subjects to provide answers for determining ability levels across concepts, a simulation approach was adopted. The simulation programme generates input vectors, mimicking the output that would typically arise from an AQG process, as outlined in Section 4.2. The simulated learner's response to a concept is determined by the current ability level associated with that concept and the question's level of difficulty. Consequently, probabilities are assigned to yield either a true or false response. In this context, true signifies a correct response, while false indicates an incorrect one. For instance, a learner possessing a relatively low ability level, like -1.45, has a higher likelihood of yielding a false response compared to a true one. This iterative process continues until the abilities of the concepts converge or the discrepancy between the prior and current abilities falls within the range of [-0.5, 0.5].

```
Concepts learner ability = 0
while learner abilities isnot converged do
    Select questions for each concept
    Generate input vectors for each concept and store them in a file
    Input file in the adaptive system
    Update learner abilities
end while
```

Figure 6: This pseudo-code represents a simulation designed for quantitative analysis. The code models an iterative process where a learner takes a test and subsequently calculates their abilities for each concept. The simulation continues until the learner's abilities for all tested concepts have converged, at which point it concludes and provides the final ability values as output.

Three concepts were chosen for the simulation: *Ajoblanco*, *Bacon Explosion*, and *Bhajji*. After five iterations, the learner's abilities are shown in Table 3,

	Ajoblanco	BaconExplosion	Bhajji
Results	11/20	11/20	17/20
Ability	0.164	0.428	0.679

Table 3: This table presents the data obtained after the simulation's execution. In the *Results* row, the traditional method for calculating test results is depicted as the ratio of correct answers to the total number of questions. Additionally, the *Ability* row displays the estimated learner abilities for each concept, derived from the adaptive system's calculations.

Comparing the learner abilities produced using IRT to the results of the test when scoring it the conventional way, that is, by taking the number of correct answers and dividing by the total number of questions, would show us how the two different scoring methods differ and by how much. In order to translate the learner's abilities into the same metric as a percentage, they should be normalised using the formula:

$$NormalisedAbility = \frac{Ability - (-4)}{4 - (-4)}$$

	Ajoblanco	BaconExplosion	Bhajji
Normalised Ability	52.05%	55.35%	58.49%
Traditional scoring	55%	55%	85%

Table 4: In this table, we are conducting a comparative analysis between the conventional scoring method and the IRT scoring method. The abilities presented in Table 3 have been standardised, and the corresponding percentages from Table 3 have been computed. This enables us to perform a detailed comparison and analysis of the two sets of values for each concept.

Lastly, the last factor to take into account is the $difficultyFactor$ as explained in Section 4.2. The $difficultyFactor$ plays a small role in calculating IRT learner abilities but does not when scoring a test using conventional methods. Table 5 shows the results for the number of difficult questions answered correctly for each concept.

	Ajoblanco	BaconExplosion	Bhajji
Difficult Questions	1/6	2/6	3/6

Table 5: This table displays the number of correct difficult questions answered by the simulation for each concept.

5.3 System performance

It is critical to test the adaptive system's performance with large data sets and analyse its behaviour under extreme conditions. This is especially important when considering the system's use in creating and processing lengthy assessments. We must consider two key aspects when constructing tests for the adaptive system: the number of concepts per test and the number of questions relevant to each concept.

As outlined in Section 4.3, every distinct concept undergoes processing via the IRT unit. The adaptive system is intended to include a maximum of five distinct concepts in every exam, although this capacity can be changed depending on the requirements of the test. As a result, running the adaptive system requires performing five IRT calculations.

The first performance test explores what would happen to the execution time of the IRT unit as the number of concepts per test increased. Five dummy tests were setup, all with a different number of concepts. To mimic reality, each concept contained three to five questions related to it in the test. In reality, it is not guaranteed that each concept will have an equal number of questions related to it. Another advantage of this approach is that it would give us an idea of roughly how many questions you could expect if you were to have a specified number of concepts. The results of the evaluation are shown in Figure 7.

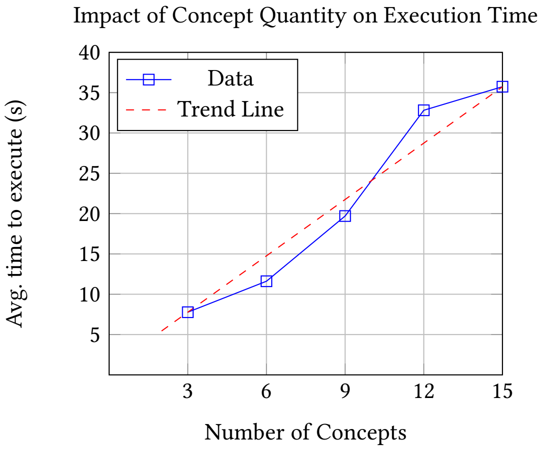

Figure 7: This graph illustrates the variation in the average execution time of the adaptive system as the number of concepts in the test increases. To generate the data points, the adaptive system is executed three times using the same number of concepts in each test, and the resulting execution times are averaged. The maximum number of concepts per test was 15.

The second performance assessment involves a comparison of execution times between creating a conventional IRT test (i.e., determining ability for an entire test) and employing our adaptive system to generate a test. Our adaptive system computes the ability for each concept, leading to an increase in test length as the number of concepts rises (assuming the number of questions per concept remains constant). In order to simulate a traditional IRT test scenario, the concept count was limited to one. This approach necessitates a single IRT calculation per test, mirroring the methodology adopted in conventional IRT tests. For a fair comparison of the two test scenarios, our adaptive system adjusts the number of concepts per test, similar to the procedure used in the initial performance evaluation. Conversely, the conventional IRT test alters the test length, approximating the average length derived from the adaptive system test with x concepts. Table 6 visually outlines this process.

Number of concepts	Average test length
3	12
6	23
9	38
12	45
15	59

Table 6: This table displays the test length based on the number of concepts in the test. During the evaluation, 3-5 questions were generated for each concept included in the test.

Similarly to the first performance test, the execution time of the conventional IRT test method was measured. The results of these tests are shown in Table 7.

Test length	Execution time (s) (Adaptive system)	Execution time (s) (Conventional IRT test)
12	7.77	1,46
23	11.61	3.00
38	19.70	2.22
45	32.82	3.06
59	35.74	3.94

Table 7: This table provides a comparison of execution times between the adaptive system and a conventional IRT assessment across different test lengths.

To visualise the speed difference between using IRT for a conventional test compared to using IRT in our adaptive system, the ratio of the execution time was displayed as a graph shown in Figure 8.

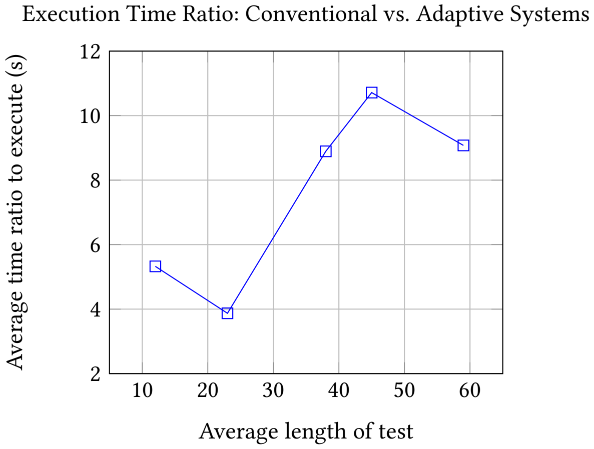

Figure 8: This graph takes the ratio of the execution time of the adaptive system and the conventional IRT assessment from Table 7.

6 DISCUSSION

In the first assessment, the accuracy of the system was put to the test through user evaluations. Table 2 presents the findings of this assessment. The majority of participants found that the adaptive system was moderately accurate in estimating their proficiency levels for each concept. This indicates that the system's estimations were not entirely off and were moving in the right direction. However, the participants reported that the system was only 33% accurate in estimating their true abilities, with 22% stating that the estimation was completely inaccurate. While the system is capable of generating precise results, it is not flawless. Its performance varies based on the context, and it requires refinement in its ability to estimate learners' abilities. To achieve more consistent and accurate results, the algorithm and methods used to compute learner abilities require improvement. One approach to enhancing the evaluation process is by recruiting more participants and altering the concepts tested to obtain a more accurate representation of the system's performance. The concepts assessed in the test were unfamiliar to the participants, which made them more susceptible to guessing. To address this issue, a guessing parameter could be incorporated into the calculation of the learner's ability.

In the second evaluation, the adaptive system's functionality was tested by comparing its scoring to traditional test scoring. The simulation results are presented in Table 3. Based on the data, *Bhajji* demonstrated the highest level of ability, resulting in the highest outcome. In contrast, *Ajoblanco* exhibited the lowest ability but achieved the same score as *BaconExplosion*, despite having a higher level of ability. The difference in ability levels between *Ajoblanco* and *BaconExplosion*, despite the same score, is due to the difficulty factor, which is highlighted in Table 5. The difficulty factor considers how well a learner answers difficult questions. *BaconExplosion* had a higher ability than *Ajoblanco* because it performed better on the difficult questions. To compare the adaptive system test scoring with traditional test scoring, ability levels were normalised to the same scale, as shown in Table 4. For *Ajoblanco* and *BaconExplosion*, the difference in scoring methods is relatively small. However, *Bhajji's* scores varied significantly, with 85% using traditional scoring and 58.49% using normalised ability. The non-linear relationship between learner ability and score is typical in IRT assessments, as illustrated in Figure 1 of Section 3. Minor adjustments in the [-2; 2] range for learner ability result in significant score changes, but beyond this range, changes in ability have less of an impact on results. *Bhajji's* score results from this non-linearity, as traditional scores imply simpler relationships with underlying ability.

The adaptive system underwent a performance evaluation that included two tests. The first test examined how the system's execution time would be affected by an increase in the number of concepts tested. The results, shown in Figure 7, indicate that the execution time directly correlates with the number of concepts, which is expected since the program is unable to process multiple concepts simultaneously. This linear increase in execution time should not be problematic for small-scale tests. However, it may pose issues for larger assessments, as discussed in Section 2.1. The second test compared the execution time of the IRT assessments of the adaptive system with conventional IRT assessments. Table 7 presents the execution times of both methods. The conventional use of IRT assessment is significantly faster than the proposed method in this research paper. Figure 8 visually portrays the correlation between the execution time ratio of conventional IRT assessment to that of the adaptive system and the average test length. It becomes evident that as the average test length spans from 25 to 45 questions, the graph exhibits a notable, predominantly linear rise. Although minor fluctuations are discernible, the overarching trend signifies that the execution time for conventional IRT assessments remains relatively stable, while the adaptive system's execution time demonstrates a linear increase. This observation underscores the significant performance advantage of systems employing conventional IRT assessments. However, systems that use conventional IRT assessments and the adaptive system serve different purposes. The conventional IRT assessment calculates the learner's overall ability throughout the test. While the adaptive system calculates the learner's ability for each concept. The manager or administrator of the adaptive system can decide on its intended use and whether they are willing to sacrifice execution time for more information. There are methods to improve the performance of the adaptive system. These include employing techniques such as multi-threading or using more efficient algorithms for IRT calculations.

7 CONCLUSIONS

The aim of this study is to create an adaptive e-learning system to address the issues of existing e-learning systems. Specifically, the failure to accommodate individual learning needs, such as addressing areas where the learner is weakest. This study introduced an adaptive e-learning system capable of assessing a learner's proficiency in different topics, with the primary aim of identifying topics in which a learner is weakest. The evaluations conducted on the adaptive e-learning system revealed valuable insights into its potential. The system has demonstrated its capability to accurately estimate learners' proficiency levels and identify their areas of weakness during testing. In comparison to conventional IRT test scoring, the system showcased its ability to provide more in-depth insights into learners' abilities, albeit with longer execution times. While the system demonstrated varying levels of accuracy in estimating learners' abilities, it also highlighted areas that require improvement. Notably, contextual factors were found to influence estimation accuracy, underscoring the need for enhanced algorithms and enhanced overall system performance, particularly when applied to large-scale assessments.

The significance of this proposed adaptive system stems in its ability to transform the type of educational content presented to students. By doing so, it overcomes the constraints of the "one-size-fits-all" approach commonly found in existing e-learning platforms. This personalised approach takes into account individual learning paces and weaknesses, offering the potential to enhance subject understanding and knowledge retention.

In Section 8.1, we introduce two additional systems, namely an Automatic Question Generator (AQG) and a Natural Language Generation Algorithm (NLGA). The significance of these systems lies in their potential for easy integration with the adaptive system, thus forming a complete educational assessment system. This integrated system is capable of dynamically generating personalised educational materials aimed at effectively identifying and addressing learners' areas of weakness. Furthermore, it can generate a structured "guide" document that specifies the exact areas on which learners should focus to improve their academic performance. This technique not only improves the adaptability of the learning experience but also provides learners with essential guidance to help them thrive in their educational pursuits.

The shift from conventional e-learning to adaptive e-learning represents a promising stride towards more effective and personalised education. The goal of meeting individualised learning needs through technology remains central to educational innovation. Despite the existing challenges and the need for adjustments, the adaptive system embodies the dynamic nature of education in an ever-changing digital era.

8 FURTHER WORK

8.1 Integrating the adaptive system

The adaptive system in this study identifies knowledge gaps in learners by processing test responses and generating a CSV file containing a learner's ability levels across various subjects (as described in Section 4.1). These inputs are obtained from an AQG, which accesses the same ontology as the adaptive system and generates questions based on CSV data containing test-specific triples (see Section 4.7). The AQG and adaptive system collaborate to dynamically create educational content that identifies a learner's weakest areas.

The CSV file from the adaptive system is also sent to a Natural Language Generation Algorithm (NLGA), which presents the AQG and adaptive system results in a learner-friendly format. The NLGA tracks CSV content, identifies concepts with the lowest learner abilities, and generates descriptive sentences explaining these concepts and the specific areas where learners struggle. The NLGA's aim is to effectively convey results and provide learners with a guiding document to enhance their understanding of weaker areas.

For a visual representation of this integrated system, please refer to Figure 10 in the Appendix. Integrating the adaptive system, AQG, and NLGA creates new possibilities for personalised and effective learning. This fully integrated system offers a glimpse into the potential future of education, where technology aligns with individual learning needs to promote understanding and development.

8.2 Development of the adaptive system

In this initial iteration of the adaptive system, there is room for multiple improvements to enhance its performance, expand functionality, and introduce additional features. Two key modifications can enhance the adaptive system's performance.

The first improvement focuses on optimising the execution time of the IRT unit. While the system performs efficiently with a small number of concepts, the execution time increases notably as the number of concepts grows, as shown in Figure 7 and discussed in Section 6. To address this challenge, a viable solution is to implement multi-threading during IRT calculations for each concept. Multi-threading uses multiple threads to manage software tasks, enabling parallel execution on a multiprocessor and thus improving performance. This approach is commonly employed in tasks like building responsive servers and various computational applications [18].

The second improvement concerns the retention of learner ability data. Currently, the adaptive system lacks the capability to preserve previous learner data when the programme is terminated and restarted. This limitation can hinder learners who wish to access their past learning abilities to evaluate progress or compare ability levels across different sessions. An effective solution is to implement memory storage for the learner's abilities. This could involve storing the learner's ability dictionary in a file within the application or site hosting the system. When the adaptive system is relaunched, users could have the option to load this file, enabling them to retrieve and review their prior ability levels.

REFERENCES

[1] Boyinbode, O., Olotu, P., and Akintola, K. Development of an ontology-based

adaptive personalized e-learning system. *Applied Computer Science 16*, 4 (2020), 64–84.

[2] Cojocariu, V.-M., Lazar, I., Nedeff, V., and Lazar, G. Swot anlysis of e-learning educational services from the perspective of their beneficiaries. *Procedia-Social and Behavioral Sciences 116* (2014), 1999–2003.

[3] Collares, C. F., and Cecilio-Fernandes, D. When i say… computerised adaptive testing. *Medical education 53*, 2 (2019), 115–116.

[4] Do, C. B., and Batzoglou, S. What is the expectation maximization algorithm? *Nature biotechnology 26*, 8 (2008), 897–899.

[5] Felder, R. M., Soloman, B. A., et al. Learning styles and strategies, 2000.

[6] Hambleton, R. K., and Van der Linden, W. J. Advances in item response theory and applications: An introduction, 1982.

[7] Hanson, B. A. Irt parameter estimation using the em algorithm, 1998.

[8] Harandi, S. R. Effects of e-learning on students' motivation. *Procedia-Social and Behavioral Sciences 181* (2015), 423–430.

[9] Karampiperis, P., and Sampson, D. Adaptive learning resources sequencing in educational hypermedia systems. *Journal of Educational Technology & Society 8*, 4 (2005), 128–147.

[10] Kulaglić, S., Mujačić, S., Serdarević, I. K., and Kasapović, S. Influence of learning styles on improving efficiency of adaptive educational hypermedia systems. In *2013 12th International conference on information technology based higher education and training (ITHET)* (2013), IEEE, pp. 1–7.

[11] Latu, E., and Chapman, E. Computerised adaptive testing. *British journal of Educational technology 33*, 5 (2002), 619–622.

[12] Livingston, S. A. Basic concepts of item response theory: A nonmathematical introduction. research memorandum. ets rm-20-06. *Educational Testing Service* (2020).

[13] Mahmud, J. Item response theory: A basic concept. *Educational Research and Reviews 12*, 5 (2017), 258–266.

[14] Meijer, R. R., and Nering, M. L. Computerized adaptive testing: Overview and introduction. *Applied psychological measurement 23*, 3 (1999), 187–194.

[15] Oproiu, G. C. A study about using e-learning platform (moodle) in university teaching process. *Procedia-Social and Behavioral Sciences 180* (2015), 426–432.

[16] Pashler, H., McDaniel, M., Rohrer, D., and Bjork, R. Learning styles: Concepts and evidence. *Psychological science in the public interest 9*, 3 (2008), 105–119.

[17] Ra, S., Shrestha, U., Khatiwada, S., Yoon, S. W., and Kwon, K. The rise of technology and impact on skills. *International Journal of Training Research 17*, sup1 (2019), 26–40.

[18] Rinard, M. Analysis of multithreaded programs. In *International Static Analysis Symposium* (2001), Springer, pp. 1–19.

[19] Surjono, H. D., et al. The design of adaptive e-learning system based on student's learning styles. *International Journal of Computer Science and Information Technologies 2*, 5 (2011), 2350–2353.

[20] Tirziu, A.-M., and Vrabie, C. Education 2.0: E-learning methods. *Procedia-Social and Behavioral Sciences 186* (2015), 376–380.

[21] Wise, S. L. Overview of practical issues in a cat program.

[22] Wise, S. L., and Kingsbury, G. G. Practical issues in developing and maintaining a computerized adaptive testing program. *Psicológica 21*, 1 (2000), 135–155.

APPENDIX

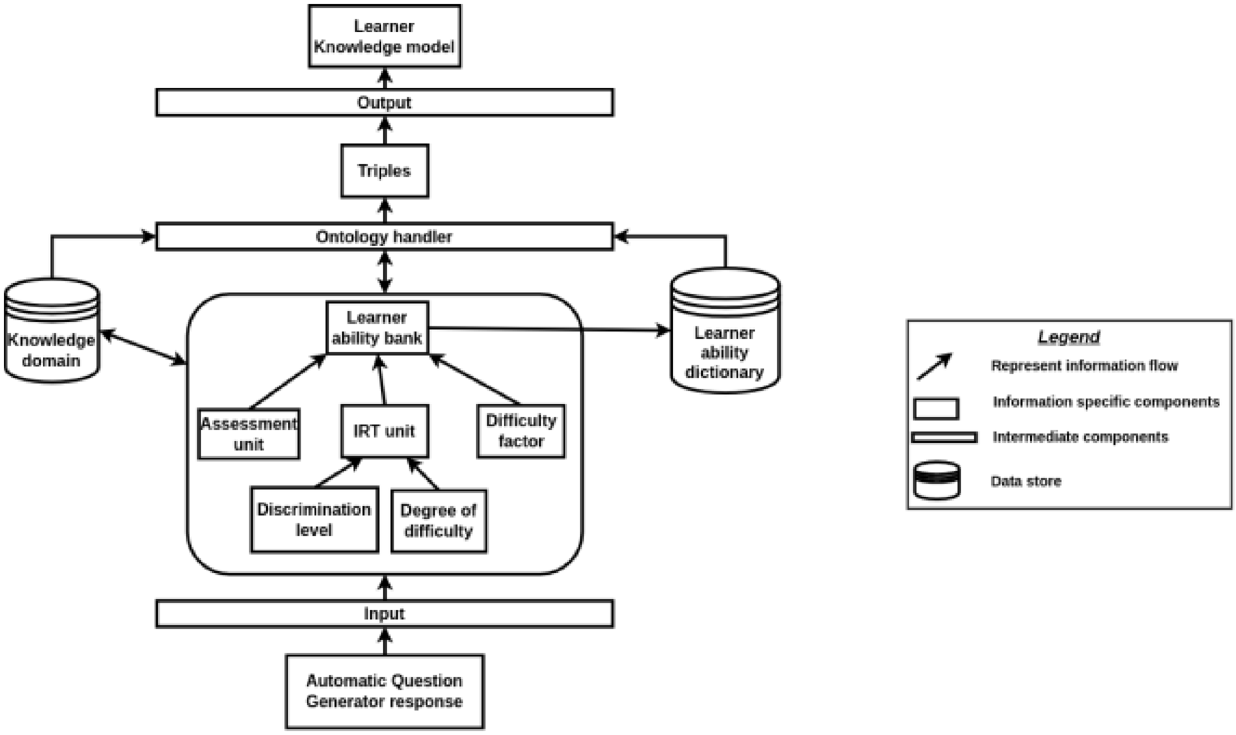

Figure 9: System Architecture. This figure presents a complete picture of the adaptive system's architecture, highlighting the interrelationships between its numerous components. This complex framework contains a total of ten interrelated elements (not including the input and output components). These elements are connected by directional arrows. Based on their functions and responsibilities, the components can be classified into three major types. Regular rectangles represent components responsible for calculating and storing specific information. Elongated rectangles represent intermediate components that manage, manipulate, or facilitate the transfer of information between regular components. Cylinders represent data sources that interact to retrieve or store crucial information, serving as core elements for other components.

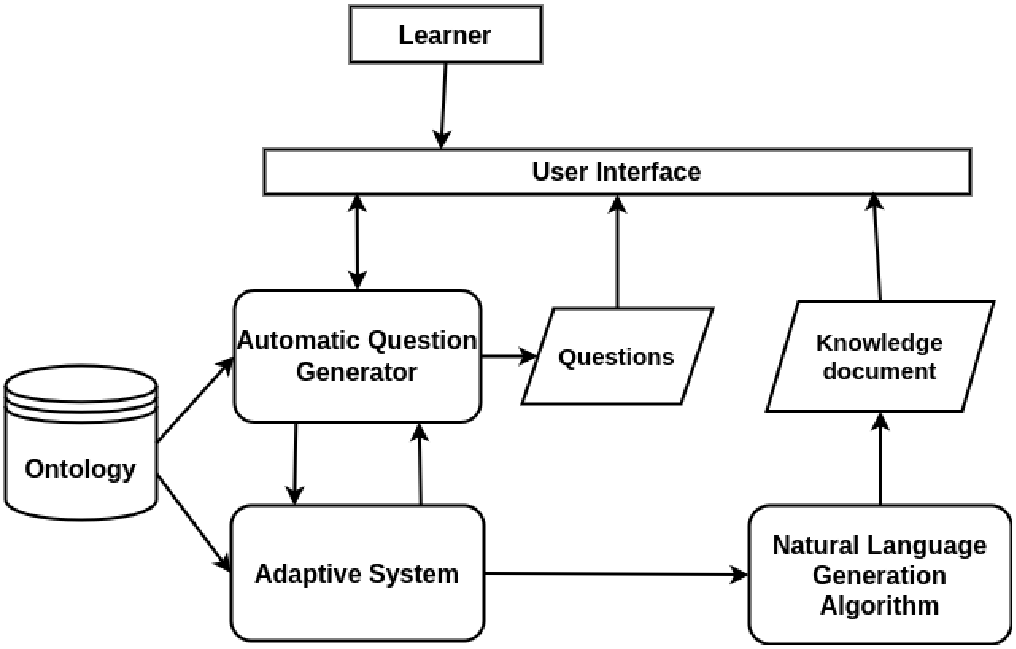

Figure 10: Full system integration. This diagram provides a visual representation of the integration among the AQG, adaptive system, and NLGA. These three components are encapsulated within rounded rectangles. Rhombus shapes represent any documents generated by the system and presented to the user. The directional arrows denote the interactions between these components. Notably, only the AQG and NLGA modules are responsible for displaying information on the user interface, while the adaptive system remains opaque to the user, functioning as a "black box".