

Article

Diag-Skills: A Diagnosis System Using Belief Functions and Semantic Models in ITS

Nesrine Rahmouni ^{1,*}, Domitile Lourdeaux ², Azzeddine Benabbou ^{3,*} and Tahar Bensebaa ¹¹ Computer Science Department, Université de Badji Mokhtar, Annaba 23000, Algeria; t_bensebaa@gmail.com² CNRS Heudiasyc UMR 7253, Université de Technologie de Compiègne, Alliance Sorbonne Université, 60200 Compiègne, France; domitile.lourdeaux@hds.utc.fr³ Institut de Recherche en Informatique de Toulouse, Université de Toulouse 3, 31000 Toulouse, France

* Correspondence: nesrinerahmouni.pro@gmail.com (N.R.); azzeddine.benabbou@irit.fr (A.B.); Tel.: +213-699392560 (N.R.)

Abstract: This work is related to the diagnosis process in intelligent tutoring systems (ITS). This process is usually a complex task that relies on imperfect data. Indeed, learning data may suffer from imprecision, uncertainty, and sometimes contradictions. In this paper, we propose Diag-Skills a diagnosis model that uses the theory of belief functions to capture these imperfections. The objective of this work is twofold: first, a dynamic diagnosis of the evaluated skills, then, the prediction of the state of the non-evaluated ones. We conducted two studies to evaluate the prediction precision of Diag-Skills. The evaluations showed good precision in predictions and almost perfect agreement with the instructor when the model failed to predict the effective state of the skill. Our main premise is that these results will serve as a support to the remediation and the feedbacks given to the learners by providing them a proper personalization.

Keywords: belief functions; diagnosis; technology-enhanced learning; tutoring systems; uncertainty



Citation: Rahmouni, N.; Lourdeaux, D.; Benabbou, A.; Bensebaa, T. Diag-Skills: A Diagnosis System Using Belief Functions and Semantic Models in ITS. *Appl. Sci.* **2021**, *11*, 11326. <https://doi.org/10.3390/app112311326>

Academic Editor: Enrico Vezzetti

Received: 19 October 2021
Accepted: 20 November 2021
Published: 30 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Intelligent tutoring systems (ITSs) have widely been used in education and have become one of the most popular areas of research and development [1,2]. One of the strengths of these systems is their ability to offer personalized learning, at any time, for learners from different backgrounds and with different profiles [3–6]. To be able to offer this dynamic personalization, it is necessary to capture the learner's characteristics (knowledge, abilities, needs, etc.) [7–10] which are commonly represented in a student/learner model [11,12]. However, it is difficult to identify accurately these characteristics. More specifically, it cannot be accurately said that a student has learned a concept or not. In fact, in certain cases, students can accidentally give a correct answer (a guess), or inversely, they can give a wrong answer without having any flaw in the skill (a slip). Thus, the diagnosis process often deals with uncertainty [12–14].

To take into account uncertainties, several research works use Artificial Intelligence (AI) techniques. They proved to be successful means to handle problems with incomplete and/or uncertain data [2,11]. As far as adaptive e-learning systems are concerned, fuzzy logic and Bayesian networks are some of the most used techniques to assess and diagnose student knowledge and skills [14,15]. However, their utility remains limited when it comes to combining information from different sources and capturing potential contradictions. In this work, we propose to use the theory of belief functions, also called Dempster–Shafer theory of evidence [16], which is a popular mathematical framework that deals with uncertain information. It is used in several domains such as decision making, risk and reliability evaluation, [17,18] sensor data fusion, and aggregation operators [19]. In addition, to its ability to deal with uncertain and imprecise information, this framework provides a natural and powerful way for the expression and fusion of uncertain information from heterogeneous sources and handle the existing conflict between these sources [20].

In this paper, we present a diagnosis module called Diag-Skills. This module is based on the pairing of a semantic model (domain ontology) and the theory of belief functions. Diag-Skills uses the learner's evaluation grades to determine the state of the evaluated skills and predicts the state of the non-evaluated ones. This diagnosis serves as a support to the personalization of the learning. Therefore, we raise the following research question: how to consider the uncertainty related to the evaluation grades to achieve a better diagnosis that will naturally lead to better-personalized learning?

This paper is organized as follows. In Section 2, we list some related works. In Section 3, we expose the system's architecture in which Diag-Skills is integrated. In Section 4, we present Diag-Skills and explain how belief functions are used to model uncertainty. In Section 5, we describe the experiment realized to evaluate Diag-Skills. Finally, we discuss the experimental results and present our perspectives for this work in Section 6.

2. Related Works

As stated in the introduction of this paper, the diagnosis process in ITS is often marked by uncertainty. This issue had led several researchers to consider different approaches to take this uncertainty into account, and improve the quality of the diagnosis process consequently.

Bayesian networks are one of the most widely used approaches that deal with uncertainty in the diagnosis process [10]. A Bayesian network is a directed acyclic graph in which the nodes represent the variables and the edges represent the probabilistic dependencies or causal relationships between these variables. Among the systems that use such networks, we can mention TELEOS (Technology Enhanced Learning Environment for Orthopedic Surgery) [21] that uses a Bayesian student model to diagnose the students' knowledge states and cognitive behaviors. Other systems and computing environments have used these networks in different domains such as TIDES [22], AdaptErrEx [23], and INQPRO [24]. The Bayesian networks offer a well-defined formalism that lends itself to sound probability computations of unobserved nodes from the evidence of observed nodes [25]. However, one major limitation of the use of Bayesian networks is that they need to be initially set with probabilities, and it is very hard for the domain's expert to encrypt these probabilities, when (s)he does not have enough data. Furthermore, the collection of these data is usually difficult.

Another well-established approach for reasoning with uncertainty in the diagnosis process is the use of fuzzy logic [26]. Some researchers use fuzzy logic to model uncertainty that is due to the subjectivity of some human characteristics and imprecisions of the learner's data [10]. The fuzzy logic is a generalization of the classical set theory. It introduces the notion of degree of membership which allows a condition to be in a state other than true or false. It also introduces the use of linguistic variables and the design of models that are both intelligible and runnable by computer systems. Therefore, many learning environments use this approach to model the learners' characteristics such as [5,27,28]. However, it has also some limitations. The rule base and membership functions become difficult to acquire when the domain model is complex and when there are several input variables. Moreover, when it comes to merging information coming from multiple sources, fuzzy logic fails to quantify the conflict when it exists.

Some researchers adopted a hybrid approach by combining Bayesian networks with fuzzy logic. Meltem Eryilmaz and Afaf Adabashi [2], for example, proposed a hybrid system to adaptively support students in learning an Excel course. The adaptation is achieved by modeling the students according to their knowledge level. The authors used fuzzy logic to determine the performance of the students according to their prior knowledge, and the Bayesian network to identify the state of the related topics they are ready to learn, or not, according to the evidence that comes from the fuzzy logic layer. This system benefits from the strength of both approaches; however, it remains limited when it comes to merging information and quantifying the conflict when it exists.

3. General Architecture

Our proposition is inspired by the work of [29]. Similarly, we propose a diagnosis model based on the pairing of a semantic model and the theory of belief functions. This latter is a generalization of the Bayesian and the fuzzy approaches. Thus, it brings together the strengths of both approaches in the same framework alongside other features. One of the original features of this proposal is the possibility to merge information from multiple information sources and quantify the conflict that could arise from the contradictions between these sources (e.g., evaluations, propagated beliefs). Furthermore, unlike classical probabilistic approaches, it gives a better representation of ignorance by distinguishing it from the case where the hypotheses are indeed equiprobable. Moreover, unlike [29], this model exploits the relations between the domain's skills to infer the state of those that have not been evaluated yet.

The diagnosis model that we propose is a part of a more general ITS architecture that is composed of different modules namely: the evaluation module and the orchestration module (cf. Figure 1). It is also composed of a domain model represented by an ontology. This latter allows the experts to specify the concepts of the domain and the links these concepts have with each other. This ITS is dedicated to algorithmic learning; however, since the diagnosis model is based on a generic approach, it can be applied to other learning domains.

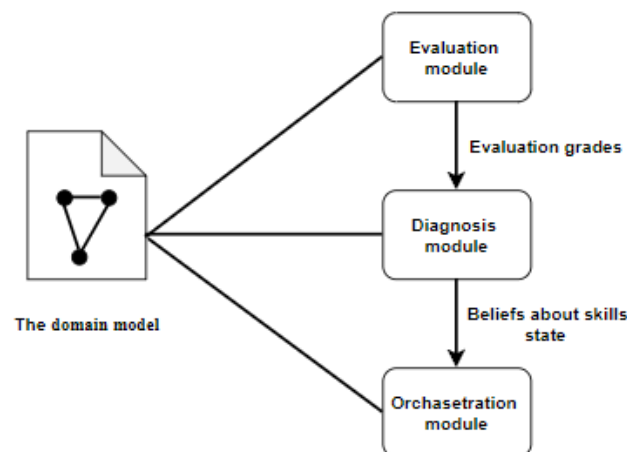


Figure 1. High-level architecture of the ITS.

In the following subsections, we present the different elements of this architecture and describe how they interact with the diagnosis module.

3.1. The Domain Model

The domain model is represented using the Algoskills ontology [30]. It gives the instructors a detailed view of different concepts in the algorithmic course. This ontology is composed of three main classes: The *Topics* class, the *Skills* class, and the *Resources* class (Figure 2).

The *Topics* class represents the hierarchy of notions in the algorithmic domain. Every notion is called a *Topic*. Each *Topic* is characterized by a noun, a definition, and one or several sub-classes. For example, the *Topic* “function” is a subclass (sub *Topic*) of the *Topic* “Algorithm structure”.

The *Skills* class represents the set of *skills* in algorithmic. It includes two types of *Skills*: disciplinary skills and transversal skills. The *Topic* class is related to disciplinary *Skills* by a semantic relation “is useful to”. E.g., the *Topic* “Recursivity” is useful to the *Skill* “identify the_recursives_structures_of_the_problem”. Finally, the *Resources* class regroup all the exercises that are used to evaluate the skills.

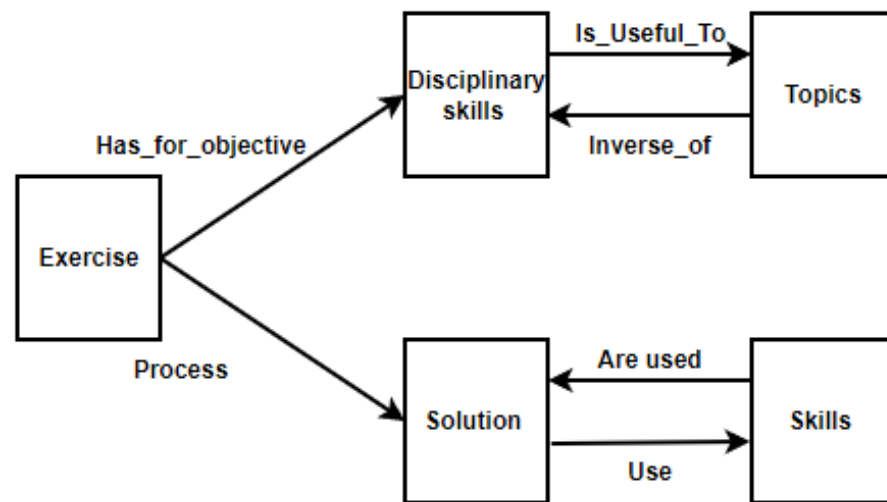


Figure 2. The Algoskills ontology.

To provide relevant feedbacks in ITS, it is necessary to know if there is any relation between the skills. For example, when a skill is acquired, it would be interesting to know which skill could be evaluated next. Furthermore, when a skill is not acquired, it would be interesting to know if this is due to a flaw in a related skill. Therefore, we propose to add the relation “Is-a-prerequisite” between the skills. Let us take, for example, two skills: S_i and S_j . S_i is a prerequisite of S_j , if the acquisition of S_i precedes the acquisition of S_j .

3.2. The Evaluation Module

This module aims to evaluate the learner’s productions. To do so, it compares these productions with the expected results. The interpretation of this comparison is expressed in numerical grades that are attributed to the corresponding skills. Once the numerical grades have been attributed, they are sent to the diagnosis module to establish a diagnosis of the learner’s skills.

3.3. The Diagnosis Module

The diagnosis model receives the evaluation grades from the evaluation module. Based on these evaluation grades, it generates beliefs about the acquisition and non-acquisition of the skills—whether they were directly evaluated or not. These beliefs are then used to express the skill’s state as a linguistic output. We defined six skill states: *acquired*, *not acquired*, *probably acquired*, *probably not acquired*, *undetermined*, and *conflictual*. These states are then transmitted to the orchestration module that takes the appropriate decisions to customize the learning session. The diagnosis model is the major contribution of this paper. We detail it in Section 4.

3.4. The Orchestration Module

This module is in charge of the orchestration of the learning sessions. It receives the states of the skills from the diagnosis module. Based on them, it generates the appropriate learning sequences. For example, if the state of a learner’s skill is *probably acquired*, the orchestration module will propose exercises to consolidate it. However, if the skill is *acquired*, the orchestration engine will tend to propose exercises in which this skill is considered as a prerequisite.

4. Diag-Skills: The Diagnosis Model

4.1. Knowledge Representation

Evaluation grades are usually based on experts' interpretations. This interpretation is sometimes subjective. Moreover, these grades can be based on an automatic evaluation process. Hence, the provided data might suffer from different kinds of imperfection:

- **Uncertainty:** learners can accidentally give a correct answer (a guess), or they can give a wrong answer without having any flaw in the skills (a slip);
- **Imprecision:** an automatic evaluation system might consider that the learner's score is in the range of 15 to 17, or produces a fuzzy linguistic variable as output (e.g., Good Performance);
- **Contradiction:** evaluation grades can be provided by different sources. Therefore, it is likely to find some inconsistencies between them;
- **Ignorance:** the system has no information at all about the skill's state.

An optimal diagnosis model has to take into account all these kinds of imperfections and be able to merge information that comes from different sources at different or similar times. To be able to do so, we propose to use the theory of belief functions. This theory allows us to represent in a single framework all kinds of data imperfection. Besides, when it comes to combining evidence from different sources, this theory provides several rules of combination that may or may not highlight the contradiction between the sources. Belief functions can be represented in different ways. We chose to represent them using a basic belief assignment (BBA) defined as a function m . This function is the mapping from elements of the power set 2^Ω onto $[0, 1]$ that satisfies: $\sum_{A \subseteq \Omega} m(A) = 1$, where $\Omega = \{\omega_1, \dots, \omega_k\}$ is a set of discrete values—called hypotheses—that can describe the state of the system.

The BBA $m(A)$ represents the belief mass attributed to A which could not be affected to a subset more specific than A . In our context, a skill is either acquired or not acquired. Therefore, $\Omega = \{acquired, \neg acquired\}$. In the remainder of this paper, we use the following notations:

- $m(a) = m(\{acquired\})$, which represents the belief mass assigned to the fact that the learner has acquired the skill;
- $m(\neg a) = m(\{\neg acquired\})$, which represents the belief mass assigned to the fact that the learner has not acquired the skill;
- $m(i) = m(\{acquired, \neg acquired\})$, which represents the belief mass assigned to the fact that the system ignores if the skill is acquired or not;
- $m(c) = m(\emptyset)$, which represents the belief mass assigned to the fact that there is a conflict resulting from certain contradictions.
- These belief masses are such that:

$$m(a) + m(\neg a) + m(i) + m(c) = 1. \quad (1)$$

4.2. The Diagnosis Process

The diagnosis process is based on five main functions as illustrated in Figure 3:

- The **Transformation** of evaluation grades into belief masses;
- The **Revision** of local beliefs;
- The **Propagation** of the revised beliefs to the neighboring skills;
- The **Merge** of the propagated beliefs with the local ones;
- The **Translation** of beliefs to a skill's state.

When a skill is evaluated, Diag-Skills receives an evaluation grade. This numerical value is transformed into mass distribution. This latter is used to perform a revision of the current mass distribution. This change in the skill's mass distribution triggers a recursive propagation to the neighboring skills. When a propagated belief is received, it is merged with the current belief thanks to the Fusion function.

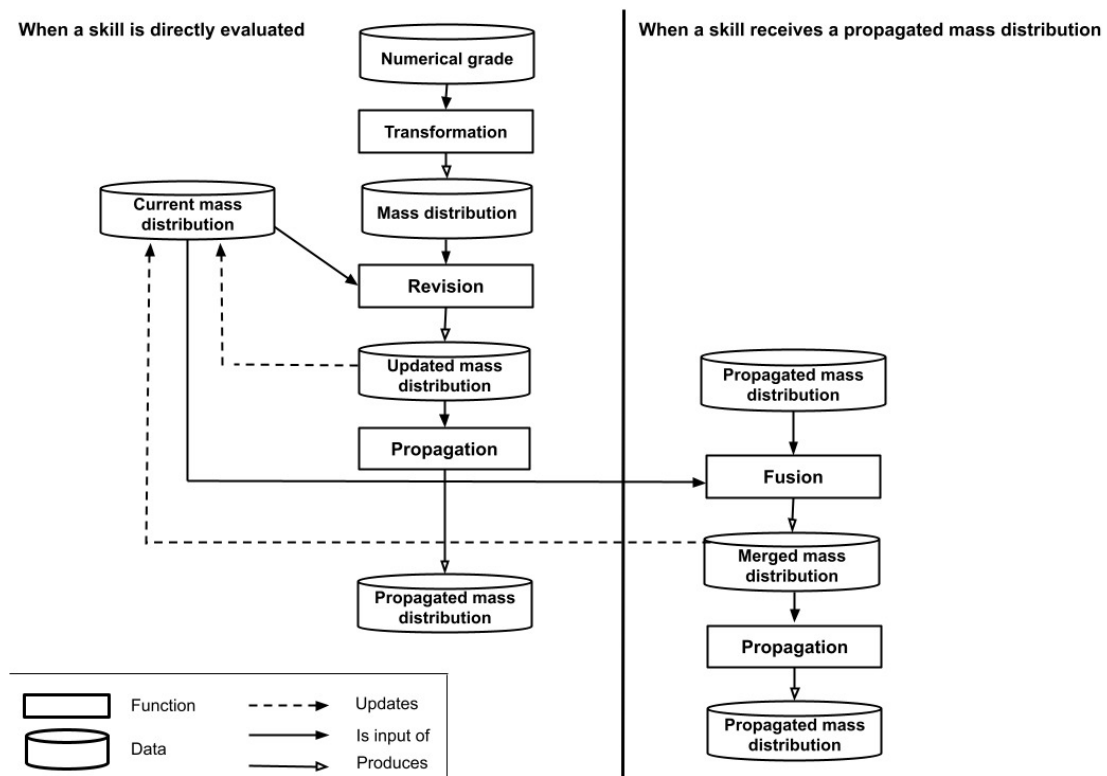


Figure 3. The diagnosis process.

4.2.1. The Transformation Function

Diag-Skills’ reasoning is based on mass distributions. It uses these masses in order to determine the state of the learner’s skills. However, as indicated in the architecture, the diagnosis module receives numerical grades from the evaluation module. These numerical values are not directly processable by Diag-Skills and need to be transformed into a distribution of belief masses. To do so, we define the transformation function $T(ne, E, \epsilon)$. The first parameter (ne) refers to the score obtained by the learner in a given assessment. It is defined in the range $[0, n]$, where 0 and n represent respectively the minimum and maximum score that the learner can get in an assessment. The second parameter (E) represents the minimum score required to validate the assessment. Finally, (ϵ) represents a range of doubt on E . Based on these parameters, this function returns distribution of belief masses (i.e., the values of $m(a)$, $m(\neg a)$, $m(i)$, and $m(c)$) such that $m(a) + m(\neg a) + m(i) + m(c) = 1$. The function is illustrated in Figure 4.

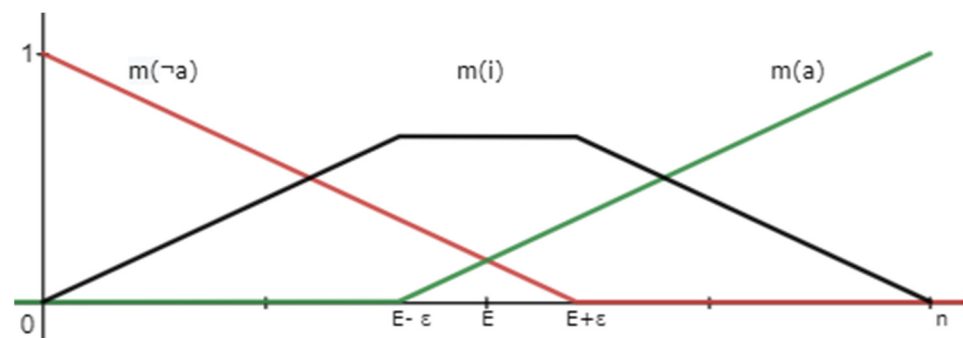


Figure 4. The transformation function.

The belief masses are defined as follows:

$$m(\neg a) = \begin{cases} -\frac{1}{E + \epsilon} x + 1, & 0 \leq x < E + \epsilon \\ 0, & E + \epsilon \leq x \end{cases}, \tag{2}$$

$$m(a) = \begin{cases} \frac{1}{n - E + \epsilon} x + \left(\frac{-E + \epsilon}{n - E + \epsilon} \right), & E - \epsilon < x \leq n \\ 0, & x \leq E - \epsilon \end{cases}, \tag{3}$$

$$m(i) = \begin{cases} 1 - m(\neg a) & 0 \leq x < E - \epsilon \\ 1 - m(\neg a) - m(a) & E - \epsilon \leq x \leq E + \epsilon \\ 1 - m(a) & E + \epsilon < x \end{cases}. \tag{4}$$

Let us take an example of a learner who got a score of 14 out of 20 on a test in which the required score to validate it is 10, with a doubt of 2 out of 20. The belief mass distribution is in this case:

$$m(a) = 0, m(\neg a) = 0.5, m(i) = 0.5, m(c) = 0.$$

4.2.2. The Revision Function

The purpose of this function is to update a skill’s state when new information arrives. It consists in taking into consideration the incoming belief and combining it with the existing one. We argue that this revision process should depend on the gap between the existing belief and the new one. We distinguish two cases. The first one refers to the case where the states of the skills are almost similar to each other (i.e., adjacent states). For example, the current state is *probably acquired* and the new state is *acquired*. The second case refers to the case where the states of the skills are considerably different from each other (i.e., not adjacent states). For example, the current state is *Probably acquired* and the new state is *not acquired*. Therefore, we define two different revision operators that we detail below.

The first revision operator is applied when the incoming skill’s state is almost similar to the existing one. In this case, we believe that it is necessary to maximize the consideration of the new information and minimize the loss of the current one. To do so, we propose to use the inner revision operator [31]. Let m_C and m_N be two mass functions over Ω and let ϕ be an inner revision operator that revises m_C using m_N such that

$$\begin{aligned} m_R(A) &= m_N(A) \phi m_C(A) \\ &= \sum_{A \subseteq B} \sigma_A(A, B) * m_A(B) \\ \sigma_A(A, B) &= \begin{cases} \frac{m_A}{Bel(B)} & Bel(B) > 0 \\ 0 & Bel(B) > 0 \text{ et } A \neq B \\ 1 & Bel(B) = 0 \text{ et } A = B \end{cases} \end{aligned} \tag{5}$$

with $Bel(B) = \sum_{CB, C \neq \emptyset} m(C)$. We apply this operator to compute the revised values of $m(a)$, $m(\neg a)$, $m(i)$, and $m(c)$:

$$\begin{aligned} m_R(a) &= \frac{m_C(a)}{m_C(a) + m_C(\neg a) + m_C(i)} * m_N(i) + m_N(a), \\ m_R(\neg a) &= \frac{m_C(\neg a)}{m_C(a) + m_C(\neg a) + m_C(i)} * m_N(i) + m_N(\neg a), \\ m_R(i) &= m_C(i) * m_N(i), \\ m_R(c) &= 1 - m_R(a) - m_R(\neg a) - m_R(i). \end{aligned} \tag{6}$$

The second revision operator is applied when the incoming skill’s state is considerably different from the current one. In this case, we believe that is necessary to attribute the same importance to current and new information. To do so, we propose to use the non-associative average. Let m_C and m_N be two mass functions, and σ a non-associative average operator that revises m_C by m_N . The revision result is defined by

$$m_R(c) = m_C(c) \sigma m_N(c). \tag{7}$$

We apply this operator to compute the revised values of $m(a)$, $m(\neg a)$, $m(i)$, and $m(c)$:

$$\begin{aligned} m_R(a) &= \frac{m_C(a)+m_N(a)}{2}, \\ m_R(\neg a) &= \frac{m_C(\neg a)+m_N(\neg a)}{2}, \\ m_R(c) &= \frac{m_C(c)+m_N(c)}{2}, \\ m_R(i) &= 1 - m_R(a) - m_R(\neg a) - m_R(c). \end{aligned} \tag{8}$$

4.2.3. The Propagation Function

The ultimate objective of Diag-Skills is to establish a global diagnosis of the learner’s skills. It is not only about diagnosing the skills that have been directly evaluated, but also about predicting the state of the skills that have not been evaluated yet. To that purpose, when a change occurs in a skill’s state, Diag-Skills triggers a top-down and a bottom-up propagation. In either case, we assume that the propagation implies a loss of information. To represent this loss of information, we use a discounting operator, which transforms each belief mass into a weaker and less informative one [16]. The discounting operation is based on a discount rate α taking values between 0 and 1. It is described as follows:

$$\begin{aligned} m^\alpha(A) &= (1 - \alpha) * m(A), \quad \forall A \subset \Omega \\ m^\alpha(\Omega) &= (1 - \alpha) * m(\Omega) + \alpha. \end{aligned} \tag{9}$$

In the following subsections, we detail how this discounting operation is adjusted to fit the particularity of each type of propagation, and we specify which belief masses are targeted by the discounting operation.

Top-Down Propagation

Top-down propagation refers to the propagation from a skill to its prerequisites. When a skill is acquired, this implies that its prerequisites are necessarily acquired. On the other hand, when it is not acquired, this does not necessarily mean that its prerequisites are not acquired. Therefore, when it comes to top-down propagation, we argue that the loss of information should not concern the belief about the acquisition of the skill (i.e., the mass $m(a)$) and should only concern the beliefs about the non-acquisition and the conflict. This loss is quantified by a discounting rate α and depends on the number of prerequisites n . Accordingly, we define the top-down discounting function \mathcal{L} that takes as input the source skill S and returns a “weaker” belief mass distribution to be propagated to its prerequisites. The function is defined as follows:

$$\begin{aligned} \mathcal{L}(S) &= (\mathcal{L}_a(S), \mathcal{L}_{\neg a}(S), \mathcal{L}_i(S), \mathcal{L}_c(S)), \\ \mathcal{L}_a(S) &= m_S(a), \\ \mathcal{L}_{\neg a}(S) &= (1 - \alpha)^n * m_S(\neg a), \\ \mathcal{L}_c(S) &= (1 - \alpha)^n * m_S(c), \\ \mathcal{L}_i(S) &= 1 - \mathcal{L}_a(S) - \mathcal{L}_{\neg a}(S) - \mathcal{L}_c(S). \end{aligned} \tag{10}$$

Bottom-Up Propagation

Bottom-up propagation refers to the propagation from a skill to the skill of which it is a prerequisite (referred to as a “parent”). When a skill is not acquired, this implies that its parent skills cannot be acquired. On the other hand, when it is acquired, this does not necessarily mean that the parent skills are acquired. Therefore, when it comes to bottom-up propagation, we argue that the loss of information should not concern the belief about the non-acquisition of the skill (i.e., the mass $m(\neg a)$) and should only concern the beliefs about the acquisition and the conflict. This loss is quantified by a discounting factor α and depends on the total number of the parent’s prerequisites n . Accordingly, we define the Bottom-up discounting function $\mathbb{Y} =$ that takes as input the source skill S and returns a “weaker” belief mass distribution to be propagated to its parents. The function is defined as follows:

$$\begin{aligned} \mathbb{Y} = (S) &= (\mathbb{Y} =_a (S), \mathbb{Y} =_{\neg a} (S), \mathbb{Y} =_i (S), \mathbb{Y} =_c (S)), \\ \mathbb{Y} =_a (S) &= (1 - \alpha)^n * m_S(a), \\ \mathbb{Y} =_{\neg a} (S) &= m_S(\neg a), \\ \mathbb{Y} =_c (S) &= (1 - \alpha)^n * m_S(c), \\ \mathbb{Y} =_i (S) &= 1 - \mathbb{Y} =_a (S) - \mathbb{Y} =_{\neg a} (S) - \mathbb{Y} =_c (S). \end{aligned} \tag{11}$$

4.2.4. Fusion Function

When propagation happens, a skill may receive a propagated belief. To determine the new skill’s state, we need to combine this propagated belief with the current one. Accordingly, we propose a fusion function that merges these two beliefs. This fusion allows Diag-Skills to enrich the diagnosis and so, improve the orchestration. For example, if the propagated belief shows that a particular skill is acquired and the current belief shows that is not, this means that an inconsistency exists in this skill’s state and that special attention should be given to this skill. To highlight this inconsistency, it is necessary to use a merge operator that quantifies the conflict between the sources. To that purpose, we propose to use the conjunctive combination rule [16,32]. To do so, we define the fusion function \mathcal{f} . This function uses the propagated belief and the current belief of the skill as the input variables and returns a merged mass distribution.

Let m_1 and m_2 be two distinct beliefs defined on the same frame of discernment. The fusion operator is defined as follows:

$$\begin{aligned} \mathcal{f}(m_1, m_2) &= m_1(A) \oplus m_2(A) \\ &= \sum_{B \cap C = A} m_1(B) * m_2(C). \end{aligned} \tag{12}$$

We apply this operator to compute the values of the mass distributions after the fusion:

$$\begin{aligned} m_f(a) &= m_1(a) * m_2(a) + m_1(i) * m_2(a) + m_2(i) * m_1(a), \\ m_f(\neg a) &= m_1(\neg a) * m_2(\neg a) + m_1(i) * m_2(\neg a) + m_2(i) * m_1(\neg a), \\ m_f(i) &= m_1(i) * m_2(i), \\ m_f(c) &= 1 - m_f(a) - m_f(\neg a) - m_f(i). \end{aligned} \tag{13}$$

Based on the propagation and fusion functions, we define the general propagation algorithm. It is a recursive process that is triggered whenever there is a change in a skill’s state. In particular, when a direct evaluation of skill is performed, top-down and bottom-up propagations are triggered. When a propagated belief is received, it is merged with the existing belief before performing a propagation in the same direction. We detail this process in the following Algorithm 1.

Algorithm 1 recursive propagation when a change occurs in a skill’s state

```

propagation (skill, flag ∈ {'TD', 'BU', 'TD-BU'})
  if (flag = 'TD-BU') then
    propagation(skill, 'BU')
    propagation(skill, 'TD')
  if (flag = "TD") then
    forall prerequisite in getPrerequisitesOf(skill) do
      propagated_belief: = top-down_propagation(skill.belief)
      prerequisite.belief: = fusion(propagated_belief, prerequisite.belief)
      propagation(prerequisite, 'TD')
  if (flag = 'BU') then
    forall parent in getParentsOf(skill) do
      propagated_belief: = bottom-up_propagation(skill.belief)
      parent.belief: = fusion(propagated_belief, parent.belief)
      propagation(parent, 'BU')

```

4.2.5. Translation Function

As indicated in the architecture, the orchestration module receives skills’ states from the diagnosis module. Thus, it is necessary to translate the mass distribution to a skill’s state before sending it to the orchestration module. To that purpose, we define the function (T), which takes the belief masses (m(a), m(¬a), m(i), m(c)) as input variables, and returns the corresponding skill’s state. Given below are the sets of rules used by this function:

- If $m(\neg a) \geq 0.75$ then the skill’s state is *not acquired*;
- If $m(a) \geq 0.75$ then the skill’s state is *acquired*;
- If $m(c) \geq 0.75$ then the skill’s state is *conflictual*;
- If $m(i) \geq 0.75$ then the skill’s state is *undetermined*;
- If $0.75 > m(a) > 0.25$ and $m(a) \geq m(\neg a)$ then the skill’s state is *probably acquired*;
- If $0.75 > m(\neg a) > 0.25$ and $m(\neg a) \geq m(a)$ then the skill’s state is *Probably not acquired*;
- If $m(a)$ and $m(\neg a) < 0.25$ and $m(i) \geq m(c)$ then the skill’s state is *undetermined*;
- If $m(a)$ and $m(\neg a) < 0.25$ and $m(c) \geq m(i)$ then the skill’s state is *conflictual*.

4.3. Implementation

Diag-Skills is developed in Python. It uses Pandas library to manipulate datasets. At this stage, the current version of Diag-Skills does not have a graphical user interface. It runs online on Google Colab Platform. The main function of the program takes as input a csv file that contains the evaluation scores of the students for each skill (cf. Table 1), and returns a csv file that includes the skills’ state after the diagnosis process (cf. Table 2).

Table 1. Example of a csv input file. The columns correspond to the domain’s skills and the rows correspond to the students. The cells’ content is the student’s numerical score for the corresponding skill. ‘NaN’ is used when a score is missing.

Students	Skill_1	Skill_2	...	Skill_n
Student_1	15	11	...	7
Student_2	11	NaN	...	0
...
Student_n	5	7	...	12

Table 2. Example of a csv output file. The columns correspond to the domain’s skills and the rows correspond to the students. The cells’ content is the skill’s state for the corresponding student.

Students	Skill_1_State	Skill_2_State	...	Skill_n_State
Student_1	<i>acquired</i>	<i>probably_acquired</i>	...	<i>not_acquired</i>
Student_2	<i>probably_acquired</i>	<i>undefined</i>	...	<i>conflictual</i>
...
Student_n	<i>not_acquired</i>	<i>not_acquired</i>	...	<i>acquired</i>

5. System's Evaluation

In the literature, several criteria have been suggested to evaluate the diagnosis approaches. One of the most widely used criteria is to assess the approach's ability to correctly predict the learner behavior based on inferred learner information. To do so, different measures can be used such as Prediction accuracy, Root mean square error, and Correlation [33]. In our evaluation, we assess the predictive power of Diag-Skills and its ability to represent the current skills of a learner. To achieve this, we suggest using the prediction accuracy measurement defined as follows:

$$\text{Prediction accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}.$$

We conducted two studies at two different universities with a total of 86 participants. The first one took place in 2020 at the Université de Technologie de Compiègne and consisted of a remote assessment in Algorithmics. The second study took place in 2021 at Université de Toulouse III and consisted of an in-situ assessment in user-interface design.

5.1. First Study

The first study was conducted at the Université de Technologie de Compiègne (UTC). The participants were first-year students in computer science engineering. The purpose was to evaluate Diag-Skills ability to predict the state of the students' programming skills based on prior evaluation results. To do so, we conducted this study with students who were enrolled in programming and algorithmic course. The objective of this latter is to give the students an introduction to algorithmic notions and teach them the basics of programming. No prior knowledge or programming skills are required to enroll in this course. This evaluation was conducted with close collaboration with the course instructor who has been teaching this course for 15 years.

5.1.1. The Domain Model

First, we designed the domain model with includes the main programming skills that are evaluated in this course. The ontology was constructed thanks to a close collaboration with the instructor. In this study, we only used a snippet of this ontology. It consists of eight classes that represent the main skills related to basic programming with tables. These classes are linked with each other by the prerequisite ("is-prerequisite") relationship. Figure 5 shows the model snippet.

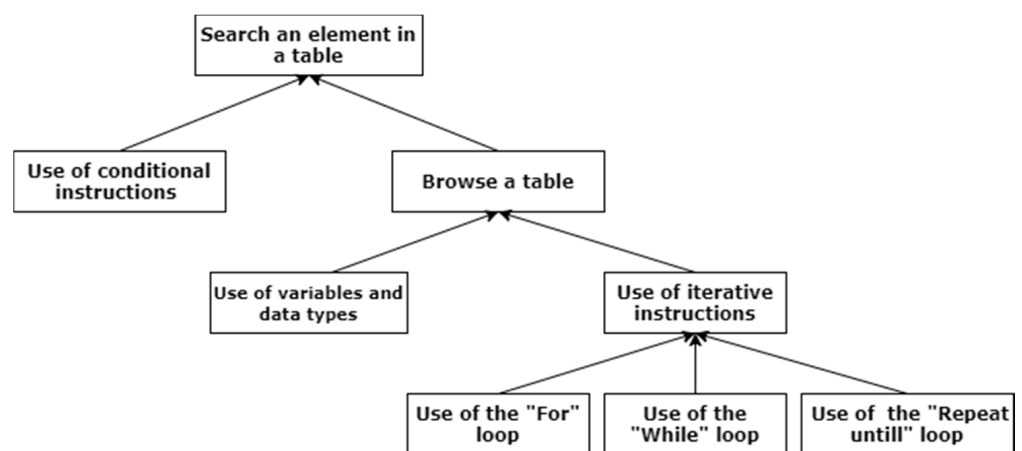


Figure 5. Snippet of the domain model.

The skills “Browse a table” and “Use of conditional instructions” are prerequisites of the skill “Search an element in a table”; the skills “Use of iterative instructions” and “Use of variables and data types” are prerequisites of the skill “Browse a table”; the skills “Use of the *for* loop”, “Use of the *while* loop” and “Use of *repeat-until* loop” are prerequisites of the skill “Use of iterative instructions”. Each skill can be in one of the following states: *acquired, not acquired, probably acquired, probably not acquired, undetermined, or conflictual*.

5.1.2. Experiment’s Process

The students received detailed lectures and practical work sessions on each notion related to the skills of the domain model. Then, they received a first test that consisted of six exercises on iterative structures (the three types of loops). Below is one example of an exercise that was given to the students in this test:

- “Write an algorithm that asks the user to enter 10 integers, then computes the number of integers that are greater than 12”.

Three weeks later, they received a second test that contained two exercises to evaluate the skills “Browse a table” and “Search an element in a table”. The respective content of these exercises consisted of the following:

- “Write an algorithm that asks the user to enter 10 numbers in a table, then computes the sum of these elements”;
- “Write an algorithm that prints the index of the first occurrence of the number 12 in a given table. The algorithm prints nothing if the number 12 is not in the table”.

The instructor assessed the answers of each student and assigned a student’s score to each skill. The purpose of this study was to evaluate Diag-Skills prediction accuracy of the state of the skill “Search an element in a table”, based on the student’s results in the first and the second test. Figure 6 details the diagnosis process.

The evaluation scores are compiled in a csv file. Then they are given as input to the main function of Diag-Skills. First, this latter takes the evaluation score of the skill “Use of iterative instructions”. This score is transformed to a belief distribution thanks to the transformation function. Then, it is propagated to the skill “Browse a table” and subsequently to the skill “Search an element in a table”. After that, Diag-Skills receives the evaluation score of the skill “Browse a table” (from the second test) and achieves a belief revision before performing a propagation. This propagated belief is merged with the belief that the system has on the state of the skill “Search an element in a table” (derived from the first test propagations). The merge’s result is translated to a state, and compared to the actual skill’s state that was derived from the second test’s score. Table 3 shows the confusion matrix that summarizes this comparison.

A total of 36 students participated in the first test, but only 26 of them attended the second one. Thus, the prediction’s accuracy was calculated using the evaluations of the students who attended both tests.

Table 3. Confusion matrix that summarizes the comparison between Diag-Skills’ predictions and the effective states of the skill “Search an element in a table”.

Diag-Skills Predictions	Acquired	Not Acquired	Probably Acquired	Probably Not Acquired
Effective Skill’s States				
<i>acquired</i>	1	0	0	0
<i>not acquired</i>	0	0	2	0
<i>probably acquired</i>	0	0	19	0
<i>probably not acquired</i>	0	0	2	2

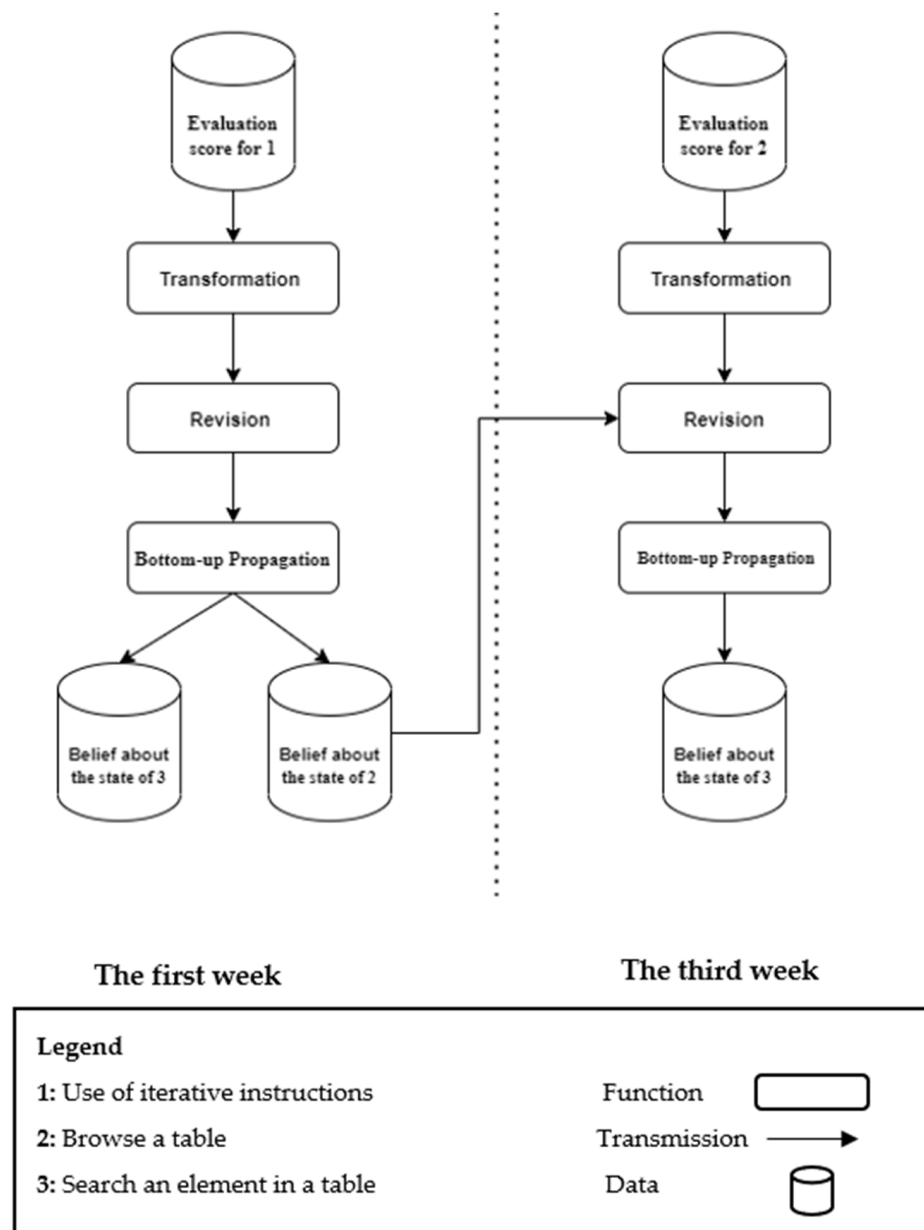


Figure 6. Prediction process of Diag-Skills in the first study.

Out of 26 predictions, 22 were correct, giving a prediction accuracy score of 84%. Since 10 students did not attend the second evaluation, we did not have the effective states of the skill “Search an element in a table”. Thus, we asked the instructor to predict the skill’s state for these 10 students and so for the four students for whom the prediction was incorrect. Then, we compared the agreement between Diag-Skills and the instructor. We used for that purpose the Kappa measure of agreement [34] defined as follows:

$$K = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{14}$$

Pr(a) is the number of observed agreements between raters, and Pr(e) is the number of agreements expected by chance. For a total of 14 predictions, the computed Kappa value was 0.85, indicating an almost perfect agreement [35] between the diagnosis and the instructor about the state of the skill “Search an element in a table”.

5.2. Second Study

In the first study, to predict the state of the targeted skill, only a bottom-up propagation was needed. To complement this study, we conducted a second one at Université de Toulouse III to evaluate the predictions based on both bottom-up and top-down propagations. This experiment involved 50 first-year students of the “Multimedia and Internet” Bachelor. They were all enrolled in the “Usability and accessibility of interfaces” course. One of the main objectives of the course is to teach students how to design prototypes of user interfaces using Adobe XD. No prior designing experience is required.

5.2.1. The Domain Model

Again, thanks to a close collaboration with the course instructor, we constructed a domain model that includes the main skills that are evaluated in this course. Figure 7 shows a snippet of this model.

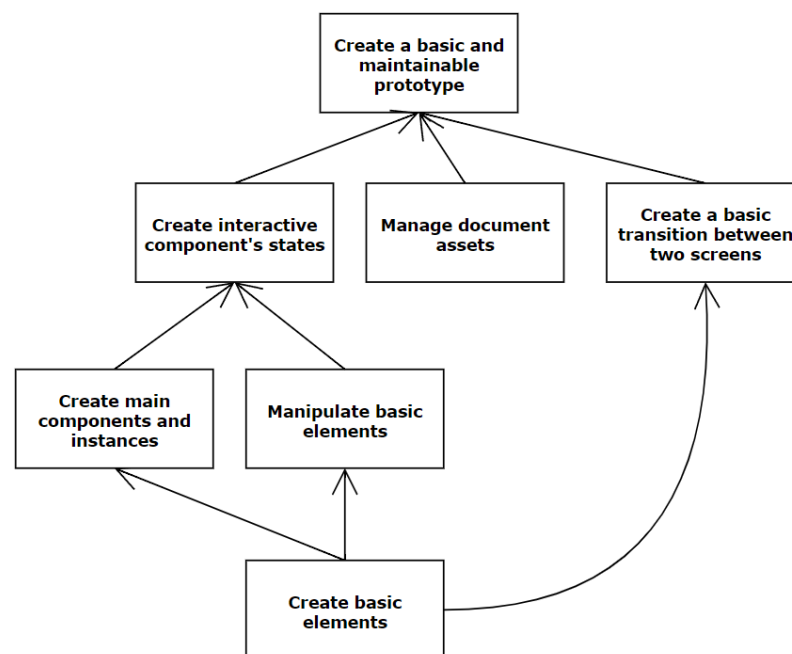


Figure 7. A snippet of the domain model.

5.2.2. Experiment Process

The students received one lecture and one tutorial class on creating prototypes. They also received exercises where they put in play all the domain’s skills. Then, they received a test where all these skills should have been applied. The objective of this test was to design a screen’s prototype containing an interactive navigation bar. The elements of this navigation bar have to change their color when the cursor hovers over them. Then, when an element is clicked on, it should send the user to the corresponding page. The expected final result was exposed to the students before they began the test. They also had step-by-step indications to achieve their tasks. Below are some examples of these indications:

- “Using the rectangle tool, add a left sidebar with a width of 350 px”;
- “Create a component containing a navigation text and a decorative side line. You will name it “Menu_item”;
- “Add a hover state to your component and modify the component and its instances so that the side line turns yellow when it is in a hover state”.

The instructor assessed each student’s production and attributed a score to each skill. The purpose of this study was to evaluate Diag-Skills’ prediction of the state of the skill: “Create interactive component’s states”. The prediction process is detailed below.

First, Diag-Skills receives the evaluation score of the skill “Manipulate basic elements”. This score is transformed to a mass distribution that is propagated recursively to the prerequisites and the parent skills. Then, Diag-Skills receives the evaluation score of the skill “Create main components and instances” and performs the same thing. After that, Diag-skills receives the evaluation score of the skill “Create a basic and maintainable prototype” and performs in particular a top-down propagation. At this stage, the final mass distribution of the state “Create interactive component’s states” should have been computed based on these propagations and the fusion function. This distribution is then translated to a skill’s state in order to be compared to the effective skill’s state. Table 4 shows the confusion matrix that summarizes this comparison.

Out of 50 predictions, 46 were correct, giving a prediction accuracy score of 92%.

Table 4. Confusion matrix that summarizes the comparison between Diag-Skills’ predictions and the effective states of the skill “Create interactive component’s states”.

Diag-Skills Predictions				
Effective Skill’s States	<i>Acquired</i>	<i>Not Acquired</i>	<i>Probably Acquired</i>	<i>Probably Not Acquired</i>
<i>acquired</i>	16	0	0	0
<i>not acquired</i>	0	6	2	0
<i>probably acquired</i>	0	0	12	0
<i>probably not acquired</i>	0	0	2	12

6. Discussion and Conclusions

As presented in the previous section, Diag-Skills showed a good prediction accuracy in the two studies. We noticed a better precision when the predictions were based on both top-down and bottom-up propagation (85% precision in the first experiment and 92% precision in the second experiment). Furthermore, we investigated the prediction when there were no effective evaluations to rely on by comparing Diag-Skills’ predictions with the instructor’s predictions. We added to these the cases of the first experiment where Diag-Skills failed to predict the correct state. The results showed that in this case, the module would at least make the same predictions as the instructor, as the Kappa measure showed an almost perfect agreement. This result suggests that Diag-Skills is capable of replicating human diagnosis with a high level of agreement.

Moreover, when analyzing the incorrect predictions, we noticed that the predicted state was always better than the effective one and never the opposite. This proves, to some extent, that Diag-Skills manages to capture the slips and the cases when there is an improvement in learning, but fails to predict the states when there is a major flaw in the skill. This should be investigated by further evaluations. Nevertheless, we noticed that in the second experiment, the mass distributions of incorrect predictions were all marked by some degree of conflict. Indeed, the conflict’s belief $m(c)$ was always greater than 0.2, indicating the presence of conflict between beliefs coming from different propagations. However, this amount of conflict was not great enough to consider that this state is conflictual. We believe, even if this amount of conflict is not great enough, that this information should be provided to the orchestration engine in order to warn it of a possible unexpected flaw in the skill.

These results suggest that Diag-Skills’ approach is a promising method for the diagnosis of learners’ skills. In particular, we showed that is possible to represent all kinds of data imperfection using a single formal framework and that is possible to highlight inconsistencies when they are relevant using this same framework. This has direct implications on the orchestration process in ITS as this information facilitates the identification of the skills that require additional evaluations. At a second level, it also has implications on the evaluation process in ITS, as this could indicate that the assessor (a computer system or an instructor) may have made inconsistent assessments and that the evaluation process should be revised.

Aside from the contributions, we would like to highlight some limitations. First, at this stage, the diagnosis relies on evaluation scores to determine the skills' states. It also depends to some extent on the number of previous evaluations. However, it does not explicitly take into consideration the amount of time between two successive evaluations nor the time between the course and the evaluation. Future work should address how this parameter can be integrated into the revision process. Also, in the translation function, we set the threshold beyond which a skill's state is considered to be conflictual to 0.75 (similarly to other states). However, the experiments showed that in reality, the states were conflictual even when the mass related to the conflict was much smaller than 0.75. Further investigations should be conducted in order to clarify if this threshold should be adjusted, and/or if the fusion with conflict should be revised. Finally, if Diag-Skills has shown good results on an absolute basis, it remains necessary to compare it empirically with other approaches.

Author Contributions: Conceptualization, N.R., D.L. and A.B.; methodology, N.R.; software, N.R.; validation, A.B., D.L.; formal analysis, N.R. and A.B.; investigation, N.R.; resources, N.R., D.L. and A.B.; data curation, N.R., D.L. and A.B.; writing—original draft preparation, N.R.; writing—review and editing, N.R. and A.B.; visualization, N.R.; supervision, D.L., A.B. and T.B.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mahmoud, A.F.A.; Belal, M.A.F.; Helmy, Y.M.K. Towards an Intelligent Tutoring System to Down Syndrome. *Int. J. Comput. Sci. Inf. Technol.* **2014**, *6*, 129–137. [[CrossRef](#)]
2. Eryilmaz, M.; Adabashi, A. Development of an intelligent tutoring system using bayesian networks and fuzzy logic for a higher student academic performance. *Appl. Sci.* **2020**, *10*, 6638. [[CrossRef](#)]
3. Chrysafiadi, K.; Virvou, M. Fuzzy logic for adaptive instruction in an e-learning environment for computer programming. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 164–177. [[CrossRef](#)]
4. Chrysafiadi, K.; Virvou, M. *Intelligent Systems Reference Library 78 Advances in Personalized Web-Based Education*; Springer: New York, NY, USA, 2015; ISBN 9783319128948.
5. Xu, D.; Wang, H.; Su, K. Intelligent student profiling with fuzzy models. In Proceedings of the 35th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, 10 January 2002. [[CrossRef](#)]
6. Gan, W.; Sun, Y.; Peng, X.; Sun, Y. Modeling learner's dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing. *Appl. Intell.* **2020**, *50*, 3894–3912. [[CrossRef](#)]
7. Shute, V.J.; Zapata-Rivera, D. Adaptive educational systems. In *Adaptive Technologies for Training and Education*; Cambridge University Press: Cambridge, UK, 2012; pp. 7–27. [[CrossRef](#)]
8. Nagatani, K.; Chen, Y.Y.; Zhang, Q.; Chen, F.; Sato, M.; Ohkuma, T. Augmenting knowledge tracing by considering forgetting behavior. In Proceedings of the WWW '19: The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3101–3107. [[CrossRef](#)]
9. Nguyen, V.A. Toward an adaptive learning system framework: Using bayesian network to manage learner model. *Int. J. Emerg. Technol. Learn.* **2012**, *7*, 38–47. [[CrossRef](#)]
10. Abyaa, A.; Khalidi Idrissi, M.; Bennani, S. Learner Modelling: Systematic Review of the Literature from the Last 5 Years. *Educ. Technol. Res. Dev.* **2019**, *67*, 1105–1143. [[CrossRef](#)]
11. Almohammadi, K.; Hagra, H.; Alghazzawi, D.; Aldabbagh, G. A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms. *J. Artif. Intell. Soft Comput. Res.* **2017**, *7*, 47–64. [[CrossRef](#)]
12. Huapaya, C.R. Proposal of fuzzy logic-based students' learning assessment model. In Proceedings of the XVIII Congreso Argentino de Ciencias de la Computación, Bahía Blanca, Argentina, 8–12 October 2012.
13. Conati, C.; Gertner, A.; Vanlehn, K. Using Bayesian networks to manage uncertainty in student modeling. *User Model. User-Adapt. Interact.* **2002**, *12*, 371–417. [[CrossRef](#)]
14. Danaparamita, M.; Lumban Gaol, F. Comparing student model accuracy with bayesian network and fuzzy logic in predicting student knowledge level. *Int. J. Multimed. Ubiquitous Eng.* **2014**, *9*, 109–120. [[CrossRef](#)]

15. Almohammadi, K. Type-2 Fuzzy Logic Based Systems for Adaptive Learning and Teaching within Intelligent E-Learning Environments. Ph.D. Thesis, University of Essex, Colchester, UK, 2016.
16. Shafer, G. *A mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
17. Deng, X.; Xiao, F.; Deng, Y. An improved distance-based total uncertainty measure in belief function theory. *Appl. Intell.* **2017**, *46*, 898–915. [[CrossRef](#)]
18. Ma, J.; Liu, W.; Miller, P.; Zhou, H. An evidential fusion approach for gender profiling. *Inf. Sci.* **2016**, *333*, 10–20. [[CrossRef](#)]
19. Jiang, W.; Wei, B.; Zhan, J.; Xie, C.; Zhou, D. A visibility graph power averaging aggregation operator: A methodology based on network analysis. *Comput. Ind. Eng.* **2016**, *101*, 260–268. [[CrossRef](#)]
20. Jiang, W.; Wei, B.; Qin, X.; Zhan, J.; Tang, Y. Sensor Data Fusion Based on a New Conflict Measure. *Math. Probl. Eng.* **2016**, *2016*, 5769061. [[CrossRef](#)]
21. Chieu, V.M.; Luengo, V.; Vadcard, L.; Tonetti, J. Student modeling in orthopedic surgery training: Exploiting symbiosis between temporal Bayesian networks and fine-grained didactic analysis. *Int. J. Artif. Intell. Educ.* **2010**, *20*, 269–301. [[CrossRef](#)]
22. Liu, C.-H. Using Bayesian networks for student modeling. In *Agent-Based Tutoring Systems by Cognitive and Affective Modeling*; IGI Global: Hershey, PA, USA, 2008; pp. 97–113. [[CrossRef](#)]
23. Goguadze, G.; Sosnovsky, S.; Isotani, S.; McLaren, B.M. Towards a Bayesian student model for detecting decimal misconceptions. In Proceedings of the 19th International Conference on Computers in Education, ICCE 2011, Chiang Mai, Thailand, 28 November–2 December 2011; pp. 34–41.
24. Ting, C.Y.; Phon-Amnuaisuk, S. Properties of Bayesian student model for INQPRO. *Appl. Intell.* **2012**, *36*, 391–406. [[CrossRef](#)]
25. Desmarais, M.C.; Baker, R.S.J.D. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Model. User-Adapt. Interact.* **2012**, *22*, 9–38. [[CrossRef](#)]
26. Zadeh, L.A. Fuzzy logic = computing with words. *IEEE Trans. Fuzzy Syst.* **1996**, *4*, 103–111. [[CrossRef](#)]
27. Tsaganou, G.; Grigoriadou, M.; Cavoura, T.; Koutra, D. Evaluating an intelligent diagnosis system of historical text comprehension. *Expert Syst. Appl.* **2003**, *25*, 493–502. [[CrossRef](#)]
28. Sani, S.M.; Aris, T.N.M.; Mustapha, N.; Sulaiman, M.N. A fuzzy logic approach to manage uncertainty and improve the prediction accuracy in student model design. *J. Theor. Appl. Inf. Technol.* **2015**, *82*, 366–377.
29. Carpentier, K. Scénarisation Personnalisée Dynamique dans les Environnements Virtuels pour la Formation. Ph.D. Thesis, Université de Technologie de Compiègne, Compiègne, France, 2015.
30. Belhaoues, T.; Bensebaa, T.; Abdessemed, M.; Bey, A. AlgoSkills: An ontology of Algorithmic Skills for exercises description and organization. *J. e-Learn. Knowl. Soc.* **2016**, *12*, 77–92.
31. Ma, J.; Liu, W.; Dubois, D.; Prade, H. Revision rules in the theory of evidence. In Proceedings of the 2010 22nd IEEE International Conference on Tools with Artificial Intelligence, Arras, France, 27–29 October 2010; pp. 295–302. [[CrossRef](#)]
32. Denceux, T. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artif. Intell.* **2008**, *172*, 234–264. [[CrossRef](#)]
33. Lallé, S. Assistance à la Construction et à la Comparaison de Techniques de Diagnostic des Connaissances. Ph.D. Thesis, Université de Grenoble, Grenoble, France, 2015.
34. Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [[CrossRef](#)]
35. Branger, B. Accord Entre Observateurs: Indice kappa de Cohen. *Réseau «Sécurité Naissance—Naître Ensemble»*. Master's Thesis, Université de Grenoble, Grenoble, France, 2009.