Modeling and Intervening Across Time in Scientific Inquiry Exploratory Learning Environment

Choo-Yee Ting*, Somnuk Phon-Amnuaisukb
Faculty of Information Technology, Multimedia University, Cyberjaya, Malaysia // *cyting@mmu.edu.my // bsomnuk@mmu.edu.my

Yen-Kuan Chong
Center for Multimedia Education Development & Application, Multimedia University, Cyberjaya, Malaysia // ykchong@mmu.edu.my

ABSTRACT
This article aims at discussing how Dynamic Decision Network (DDN) can be employed to tackle the challenges in modeling temporally variable scientific inquiry skills and provision of adaptive pedagogical interventions in INQPRO, a scientific inquiry exploratory learning environment for learning O'level Physics. We begin with an overview of INQPRO and a highlight of the computer algorithm as well as the design of our proposed DDN model. We then present an instance of interactions with INQPRO to describe how the proposed model can be generated dynamically by aggregating different INQPRO Graphical User Interfaces (GUIs) in real-time basis to perform probabilistic assessments of the two scientific inquiry skills (Hypothesis Formulation \( \mathcal{H} \) and Variable Identification \( \mathcal{S} \)). In this study, we carried out a two-phase empirical evaluation to investigate the performance of the proposed DDN model in categorizing different groups of learners. The performance of the proposed DDN model is identified by its matching accuracies elicited from a total of 6 domain experts and 77 learners who participated in both evaluation phases. Based on the empirical results, we summarized that the proposed DDN model is practically sound as it has demonstrated acceptable estimation accuracies with reference to the classification results obtained from the pretest, posttest, and from domain experts.

Keywords
Dynamic Decision Networks, Scientific inquiry learning environment, Learner modeling.

Introduction
Researchers in the field of science education have been employing scientific inquiry as an instructional strategy to maximize learner's engagement and experiences during the learning processes (Frederiksen & White, 1998; Hulshof & de Jong, 2006; Linn, 2000; Pryor & Soloway, 1997; Reiser et al., 2001; Shimoda, While & Frederiksen, 2002; Veermans & van Joolingen, 2004). The importance of this instructional strategy is proven by the increasing number of computer-assisted learning environments developed recently such as the Belvedere (Frederiksen & White, 1998), BGuilLE (Pryor & Soloway, 1997), KIE (Linn, 2000), SCI-WISE (Reiser et al., 2001), SimQuest (Veermans & van Joolingen, 2004), Rashi (Dragon et al., 2006), and SmithTown (Shute & Glaser, 1990). These learning environments do not directly deliver scientific facts to learners. Instead, learners are required to actively involve in scientific inquiry processes such as evidence gathering, constructing and testing hypotheses, manipulating variables, and the like. As an attempt to maximize scientific inquiry learning experience, exploratory learning approach is often the preference when it comes to development of computer-based learning environments (Chang et al., 2003; de Jong & Van Joolingen, 1998; de Jong, 2006). The exploratory learning approach provides learners with freedom to interact with the learning environment by means of test-and-retest their idea.

To date, although there has been an attempt to integrate intelligence into scientific inquiry learning environment by employing a learner model (Meyer et al., 1999), solutions to the following challenges remain unclear. Firstly, how should a learner model be integrated into a scientific inquiry learning environment that consists of more than a single GUI and rooted in a particular instructional model? Secondly, having gathered a learner’s exploratory behaviours, how should a system effectively assess the mastery levels of scientific inquiry skills which evolve across time? Thirdly, how should a system generate tailored pedagogical interventions in a timely manner to cater for temporally variable scientific inquiry skills? These challenges are not trivial as the system has to deal with a high level of
uncertainty inherent in inferring a learner’s mastery level of scientific inquiry skills from exploratory behaviours (Schum, 1994; de Jong, 2006).

To handle the uncertainty in an efficient modeling of interaction behaviours, Bayesian Network (BN) (Pearl, 1998) has been employed as one of the solutions (Jameson, 1995). However, a BN does not provide decision making under uncertainty. As a complementary solution, a Decision Network (DN) (Howard & Matheson, 1981; Russell & Norvig, 1995), which is an extension of BN with *decision* and *utility* nodes, is proposed by researchers in the field of Artificial Intelligence in Education (e.g., Murray et al., 2004; Conati, 2002, Pek & Poh, 2005). When time is a crucial factor in learner modeling, a static DN is then extended to a Dynamic Decision Network (DDN). By employing a DDN, not only the system is able to model the variables that evolve across time, but at the same time provides tailored feedback in a timely manner. A study by Murray et al. (2004) has shown that the decision-theoretic approach outperforms Fixed-Policy Tutor in selecting the optimal tutorial action. Here, we review instances of decision-theoretic Intelligent Tutoring Systems (ITSs) such as DT Tutor (Murray et al., 2004), CAPIT (Mayo & Mitrovic, 2001), Prime Climb (Conati, 2002), and iTutor (Pek & Poh, 2005), and subsequently highlight the major differences between their work and ours.

DT Tutor is an ITS designed for two domains, namely, the *Calculus-related rate problems* and *Elementary reading*. DT Tutor takes into consideration of not only the learner’s goals, focus of attention, and affective state, but also its objectives to provide optimal pedagogical decision. DT Tutor employs a DDN for selecting tutorial actions while assisting a learner with a task. DT Tutor was evaluated to verify if the selected tutorial action is rational and fast enough under a variety of tutorial conditions through an extensive simulation. Both DT Tutor and our work decide optimal tutorial action with maximum expected utility and exploit temporal learner properties. Although both DT Tutor and our work rely on DDN to model and intervene under uncertainty, our work differ from DT Tutor in several ways. First, a DDN is leveraged by DT Tutor to predict multiple steps ahead, while ours do not. The difference is largely because of the nature of INQPRO (this research work). INQPRO is based on exploratory learning approach (de Jong, 2006) which consists of six different interfaces. Each interface in INQPRO has a DN associated with it. Considering the number of nodes in each DN and all the possible navigation paths (state space = $5^n$ with $n \in \{\text{Integer} > 0\}$), predicting multiple steps ahead can easily be computationally intractable. Second, the emphasis of DT Tutor is studying how to recognize the evolving learner’s affective states, and focus of attention; while our work focuses on both how to recognize learner’s temporally variable scientific inquiry skills and on how learner assistance can be timely and appropriately applied within the decision-theoretic framework. Instead of merely developing a component for optimal pedagogical intervention selection mechanism, our work involves developing the interface component and the inference engine as well as their systematic integration into INQPRO. Third, the effectiveness of the DT Tutor’s tutorial action section engine is evaluated via an extensive simulation. INQPRO, however, is evaluated and validated via simulation, human learners, and domain experts.

Prime Climb is an ITS that enhances a learner’s acquisition on Prime numbers. By interacting with the learner, the *Intelligent Pedagogical Agent* monitor the learner’s emotions and generate appropriate interventions aiming at achieving the best tradeoff between the learner’s learning and engagement during their interactions with Prime Climb. To achieve this objective, the intervention is decided by maximizing the expected utility of all available decision alternatives. The DDN employed in Prime Climb is closely related to that implemented by DT Tutor, where the ‘intervening’ type (Jensen, 2002) of DDN is employed. The difference between the DDN employed by DT Tutor and that of Prime Climb is the number of *look-ahead* time-slices generated. Different from Prime Climb, however, there is no *look-ahead* time-slice due to the computational complexity in evaluating the DNs.

CAPIT, a Constraint-Based Tutor for teaching capitalization and punctuation, is an ITS that leverages decision-theoretic approach to decision making. The major difference between CAPIT, DT Tutor, Prime Climb, and our work is that CAPIT separates the learner modeling portion from the tutorial action selection engine. It has a BN that performs assessment on a learner’s knowledge states. Based on the outcome probabilities computed by the network, they are then multiplied with their associated utilities to determine which decision alternative has the maximum expected utility. Such calculation, however, is performed outside the network itself. Performing calculation of expected utilities outside a BN is deemed to potentially faster inference as *decision* and *utility* nodes are removed. According to Murray et al. (2004), such potential speedup of inference might resort to two drawbacks. First, there is a possibility that less obvious decision alternatives are not taken into consideration. Second, computation of expected utility values for each decision alternative might not be done accurately.
iTutor employs a DDN to pre-compute a tutoring policy such that the tutoring action that maximizes the expected utility. The optimal decision may belong to one of the three main categories: (i) assessing the learner’s level of knowledge mastery, (ii) presenting a lesson, and (iii) determining the learner’s readiness to leave the tutoring session. If the decision is to assess the student’s mastery, a challenging item will be selected. If the decision is to present a lesson, iTutor will select a suitable instruction from its domain knowledge database. While iTutor focuses more towards the physics concepts acquisition, our work focuses on mastering of scientific inquiry skills. Besides, rather than pre-computing a tutoring policy, our work focuses on localized and fading support as the learner interact with INQPRO. We shall further discuss the localized and fading support in a later part of this article.

The ITSs mentioned above share a common feature: the DDNs employed are having a set of nodes that are identical for all the \( n \) time-slices. Different from our work, the DDNs employed by these ITSs do not contain static node. Such approach, however, is not applicable generally. For instance in study, the INQPRO consists of six GUIs and there is a DN associated to each GUI. Thus, the main challenge in this study is to identify a sound DDN model that is able to handle the time factor gracefully without affecting its accuracy in assessing learner’s scientific inquiry skills.

In the rest of this article, we shall firstly provide an overview of DDN and the INQPRO learning environment. We then proceed with the discussion on the detailed design of our proposed DDN model. To provide a better understanding of how the proposed DDN model functions, an illustration on the probabilistic assessment of scientific inquiry skills and generating tailored interventions is given. Lastly, we shall discuss the empirical results from the evaluation of the proposed DDN through field tests administrated on human learners and classifications elicited by human experts.

**Dynamic Decision Network**

Figure 1(a) depicts a BN with three chance nodes while 1(b) depicts the extension of BN into a DN by adding utility (\( Y \)) and decision (\( A \)) nodes. A DN, which is also known as Influence Diagram (Howard and Matheson, 1981; Jensen, 2002), is extended from a BN to allow decision making under uncertainty. Similar to a BN, the formalism of DN allows encoding for probability distribution over a set of random variables. On top of that, it provides the capability to present decision alternatives that an agent can take together with the utilities of the uncertain outcomes. The qualitative perspective of a DN is a graphical representation of the variables that takes the form of a Directed Acyclic Graph (DAG).

The three types of node found in a DN are the chance node, decision node, and utility node. The chance node in DN is similar to that in BN. It represents a random variable and usually represented using a circle (nodes \( X_1, X_2, X_3 \) in Figure 1). It has an associated Conditional Probability Table (CPT), giving the probability of the variable given its parents. The parent nodes of a chance node can be both chance nodes and decision nodes. The arcs between the chance nodes encode the interdependences among the related random variables. The second type of node is the decision node (node \( A \) in Figure 1(b)). It represents all the available decision alternatives being taken at a particular time point and usually depicted as a rectangular shape. The values of a decision node represent the decision alternatives to be chosen between. To decide among the decision alternatives, the expected utility of each alternative is calculated by summing the utilities of all possible outcomes weighted by the probabilities of those outcomes:

\[
EV(A|a) = \sum_i P(\text{Outcome}_i(A)|a) \cdot U(\text{Outcome}_i(A))
\]

(Eq. 1)
From Equation (1), action $A$ has possible outcome states $\text{Outcome}_i(A)$, $\alpha$ is the current knowledge (evidence) of agent. The agent assigns probability $P(\text{Outcome}_i(A) | \alpha(A))$ prior to execution. $\text{Outcome}_i(A)$ is the proposition that action $A$ is executed in the current state. Following the principle of Maximum Expected Utility (MEU), a rational agent should then select an action that maximizes the agent’s expected utility. That is, to maximize Equation 1.

A decision node can have both chance nodes and other decision nodes to be the parent nodes. Having another decision node as parent node, the network is actually modeling a sequence of decisions to be taken. The third type of node is the utility node $Y$. It highlights the desirability of the consequences that may arise from the various decisions. Utility nodes are depicted as a diamond. It can have both chance nodes and decision nodes as parent nodes. The incoming arc of a utility node represents an influence on preference. However, there is no outgoing arc from utility nodes. The arc directed from a decision node to a chance node represent an influence on random variable exerted through an intervention from the decision alternatives at hand.

A DN in Figure 1(b) can be extended with a temporal dimension resulting in a DDN (Figure 2). A DDN is a graphical data structure that models the state of the world over time and typically represents a particular number of connected time-slices. The relationships between variables at successive time steps are represented by temporal arcs. The variables that evolve across time are considered as dynamic nodes while variables that are instantiated in a particular time-slice only are known as the temporal nodes (Schafer & Weyrath, 1997). In some instances of DDN, there are variables that do not evolve across time. Such variables are commonly expressed as the Static nodes. The DDN structure employed in this study does not have temporal arcs that span more than a single time step. This is an instance of the Markov Assumption, which the state of the world at a particular time depends only on the previous state and any decision taken in it. The relationships between the variables are quantified by the conditional probability distribution associated with each node. Once the DDN is defined, the agent’s current beliefs about the world are set as evidence in the time-slice $t_i$. By updating the DDN, the expected utility of performing each possible decision can be calculated and the agent can select the decision alternative that has the highest expected utility.

In this study, the DDN created is following the Markov assumption that the state of the world at a particular time depends solely on the immediate previous state and any action taken in it. Thus, there are no arcs that span more than a single time step. In the subsequent sections, we describe in detail the sample assessment of learner’s hypothesis formulation and variable identification skills using a scenario.

**An Overview of INQPRO Learning Environment**

INQPRO is a computer-based scientific inquiry learning environment for enhancing scientific inquiry skills acquisition. The development of INQPRO is rooted in Scientific Inquiry Exploratory Learning Model (Ting & M.Y. Arshad, 2000) (Figure 3). This model was employed mainly because it emphasizes on the explicit instructional steps that form the basis of the INQPRO development. The GUIs in INQPRO consist of the Scenario (Sce), Hypothesis Visualization (Vz), Verification (Vf), Formula (Fe), Simulation Experiment (Ex), Data Comparison (Dc), and

---

Figure 2. A general three time-slices DDN. In this study, this type of DDN has been integrated and evaluated in the INQPRO system. There is no temporal dependency between the decision nodes.
Feedback (Fz). The GUIs allow the scientific inquiry skills (Hypothesis Formulation and Testing $H$ and Variables Identification $\varsigma$) to be assessed separately (within a particular GUI) or as a whole (overall acquisition level after $n$ times of interactions). By actively interacting with the GUIs and Animated Pedagogical Agent (hereafter “Agent”), learners are ultimately expected to command two scientific inquiry skills: $H$ (node $H$ in Figure 4) and $\varsigma$ (node $V$ in Figure 4). Since $H$ and $\varsigma$ influenced each other, we introduced $K$ (node $K$ in Figure 4) during the modeling phase to represent the overall acquisition of both $H$ and $\varsigma$. This would allow us to map $K$ with the results of pretest and posttest, which formed the basis for computing the matching accuracies. To maximize learning experiences throughout the learning process, we adopted exploratory learning approach to furnish learners with the freedom to explore the domain concepts through manipulation of the scientific inquiry skills. This can be achieved by integrating rich interactive learner control components such as the drop down box, list box, track bar, and drag-drop objects in the GUIs.

![Diagram](image)

**Figure 3. The Scientific Inquiry Learning Model.** The names of INQPRO’s GUIs are given on the left pane. Each GUI has its own DN.

Figure 4 depicts the high-level presentation of INQPRO’s DNs. The network is categorized into four sub-networks with each sub-network consisting of either observable or non observable nodes. The observable nodes (nodes with prefix of $SA$, $SQ$, $AQ$, $t$) are nodes to be instantiated in light of evidence. In this context, learner’s interactions such as the keystrokes, mouse-click, drag-and-drop, and the typing events are instances of evidence. This kind of activities provides an indicator to how a learner reasons and making sense of $H$ and $\varsigma$. Examples of observable nodes in Figure 5(b) are nodes $AQ_{\text{Concept}}$, $SQ_{\text{Definition}}$, and $AQ_{\text{Scenario}}$. The $SQ_{\text{Definition}}$ for instance, aims to capture the frequency of questions asked by a learner. Once evidence is obtained and corresponding nodes are instantiated, interpreting this information into a DN allows INQPRO to suggest the mastery levels of $K$, $H$ and $\varsigma$ and subsequently trigger the Agent to intervene. Thus, the more learner interactions are captured, the higher the bandwidth (VanLehn & Martin, 1988). This subsequently helps reduce the uncertainty in modeling the mastery levels of $K$, $H$ and $\varsigma$ as they interact with INQPRO. Conversely, the non observable nodes are nodes that cannot be instantiated directly; they can only be inferred once observable nodes are instantiated and DN is updated. Examples of non observable nodes in Figure 5(b) are the nodes Hypothesis and Variable. Each DN consists of at least one decision node $A$ (nodes prefix with $AA$). The decision node in the Scientific Inquiry subnetwork for instance is meant for the Agent to provide tailored pedagogical intervention to foster the acquisition of $H$ and $\varsigma$. 

243
Figure 4. High-level presentation for general DN in INQPRO. Each DN contains four subnetworks namely the Domain Knowledge subnetwork, Scientific Inquiry subnetwork, Interface Interaction subnetwork, and self-Explain subnetwork.

To enable the modeling of a learner’s $H$ and $\zeta$ at a particular GUI, a DN needs to perform both diagnostic and predictive reasoning. By performing diagnostic reasoning, a learner’s $H$ can be inferred from the following evidences: (i) the correctness of hypothesis statement (node $SA_{\text{HypoStruct}}$), (ii) the correctness of variable relationships statement (node $SA_{\text{HypoRelation}}$), and (iii) whether or not hint is requested from the agent (node $SA_{\text{AskHypo}}$). The predictive reasoning, conversely, offers an indirect assessment through the propagation of probability from nodes $H$ and $V$ to node $K$. Detailed explanations regarding the design and integration of nodes into the GUlIs can be found in our previous work (Ting et al., 2006).

In the next section, we shall provide a high-level illustration and discussion of two GUlIs from the INQPRO learning environment namely the Scenario and Hypothesis Visualization. We present how the corresponding DNs is integrated into the GUI to allow reasoning and intervening under uncertainty.

The Scenario GUI

Figure 5 depicts the Scenario GUI of INQPRO. It is designed by taking the first two phases (Scenario Presentation and Hypothesis Formation, Variables and Relationship Identification) from the learning model (Figure 3). Upon logging into the Scenario GUI, learners are requested to select and read a scenario (Figure 5-@). Once a scenario is selected, a three dimensional computer animation is presented to the learner. The computer simulation acts as an advance organizer (Ausubel, 1968) aiming at presenting a global presentation of knowledge to be learned and foster the acquisition of scientific inquiry skills. Having read the scenario, learners then proceed to the Hypothesis section (Figure 5-@) to construct a hypothesis statement that well describe the scenario. The learner will then be requested to identify the different types of variables in the Variable section (Figure 5-@). INQPRO does not require learners to draw graph, as drawing of a graph by itself is not a trivial task and has been the object of instruction in itself (Karavavidis et al., 2003). Therefore, by selecting suitable answers for x-axis, y-axis and typing in data sets, graph will be plotted based on the relationship of variables as reflected through the hypothesis formulated. Thus, different hypotheses might result in different patterns of formulated graphs.
The Hypothesis Visualization GUI

Figure 6(b) depicts the DN for Hypotheses Visualization interface. To infer the degree to which a learner is considered to have mastered the variables (node $\text{Variable}_{Vz}$) and hypothesis (node $\text{Hypothesis}_{Vz}$) relies on whether or not s/he is able to analyze the graph (node $\text{AnalyzeGraph}$) and has understood the purpose of simulation (node $\text{UnderstandAnimation}$). However, both $\text{AnalyzeGraph}$ and $\text{UnderstandingAnimation}$ cannot be observed directly from the interface. To obtain the posterior probability values for these two nodes, the network performs diagnostic reasoning given the instantiation of evidential nodes (nodes $\text{AQ}_{\text{MassAni}}$, $\text{SA}_{\text{PlayAni}}$, $\text{SA}_{\text{DragMass}}$, $\text{SA}_{\text{ViewGraph}}$, $\text{SA}_{\text{CompareGraph}}$, and $\text{AQ}_{\text{CompareGraph}}$).

In the next section, we shall present the description of a high-level presentation of three time-slices DDN and how the DDN is generated when learners navigate from one interface to another.
A DDN Approach for INQPRO

In this study, we employed DDN to tackle the challenges in assessing temporally variable learner’s scientific inquiry skills for three main reasons. Firstly, a learner’s scientific inquiry skills evolve across time, thus capturing the dependencies between the temporally variable skills is crucial. Secondly, freedom to navigate from one GUI to another introduces a complexity in predetermining a DDN. A predetermined DDN can easily become computationally intractable as it exhibits $5^n$ state spaces (combination of different navigation paths) with $n \in \{\text{positive integer}\}$. Thirdly, if a static DN is employed, the interpretation of new evidences will lead to the reinterpretation of previous evidences. If past learning experiences are not accounted, there is a tendency for INQPRO to inaccurately model the mastery level of scientific inquiry skills and consequently the interventions generated are not tailored to the learner. In order to overcome this drawback, a DDN must be employed instead of a static DN.

Classification of Nodes in DDN

As depicted in Figure 7, time-slice $t_i$ represents the current INQPRO interface accessed by a learner while time-slice $t_{i-1}$ describes the immediate previous state. To describe the immediate subsequent interface accessed, the time-slice $t_{i+1}$ is used. These time-slices are interconnected by temporal relations, which are illustrated by the arcs joining variables that evolve over time.

Figure 7 depicts the high-level presentation of the proposed DDN structure employed in this study. The nodes in the proposed DDN structure are categorized into dynamic nodes, temporary nodes, static nodes, decision nodes, and utility nodes. These four types of nodes share a common format $\text{gui}^{\text{gui}}$ with $\text{gui} \in \{\text{All INQPRO GUIs}\}$, while $n$ denotes the frequency a particular GUI is accessed. As an example, given that a learner who accesses the Scenario GUI for twice, then $V_{\text{gui}}^{\text{gui}_1}$ represents the first access of Scenario GUI while $V_{\text{gui}}^{\text{gui}_2}$ represents the second time the similar GUI is accessed.

In this study, one of the main challenges in this study is how to model a learner’s $H$, $\zeta$, and $K$ that evolve across time. Thus, classifying the nodes $H_n^{\text{gui}_1}$, $V_n^{\text{gui}_1}$, and $K_n^{\text{gui}_1}$ into dynamic nodes is appropriate as these nodes are meant for modeling the way the mastery level of $K$, $H$, and $\zeta$ evolve from one time-slice to another. The temporal dependencies are expressed by the arcs (dotted lines) directed between the time-slices (e.g. $H_n^{\text{gui}_1} \rightarrow H_n^{\text{gui}_2}$) in Figure 7. The temporal arcs allow INQPRO to predict the acquisition levels of $H$, $\zeta$, and $K$ at the current time-slice ($t_i$) based on previous experiences instantiated as evidences in previous time-slices ($t_{i-1}$).
The second type of nodes that exist in a particular time-slice only is categorized as the Temporary nodes. These nodes reside in the Interface Interaction subnetwork (Figure 4). Example of nodes that categorized as temporal nodes are the Agent Intervention Nodes (nodes with prefix AQ_) and Learner Exploration Nodes (nodes with prefix SQ_, SA_) The unique property of temporal nodes is that this type of nodes allow the instantiation of evidences at a particular GUI in time slice $t_n$, not to have the similar evidences instantiation at the similar GUI in time slice $t$. For instance, presumably in time-slice $t_n$, a learner has typed in a wrong hypothesis statement resulting in the node $SA\_HypoRelation$ and $SA\_HypoStruct$ to be instantiated to non-mastery. However, revising and refining the hypothesis statement in time-slice $t$ has subsequently resulting in the nodes instantiated to mastery.

Apart from the dynamic nodes and temporal nodes, the nodes $S^H, S^F$, and $S^V$ in Figure 7 are categorized as static nodes. The probabilities of a static node may change but it is not a variable that evolves significantly with respect to each time-slice. Conversely, due to the conditional effect of dynamic nodes (nodes $P^{AQ}_t$, $P^{SQ}_t$, $P^{SA}_t$) on the Static nodes (nodes $S^H, S^F$, and $S^V$), a consistent rather than drastic change of probability propagation is observed. This conditional effect provides an “accumulative” probabilistic assessment of $H, z, \zeta$ and $K$ after a series of interactions with INQPRO during the learning process. Such mechanism has provided the DDN with the functionality to remember and consider the mastery level of the immediate previous time-slice when reasoning the mastery levels of $H, \zeta, \zeta$, and $K$ in current time-slices.

A learner who revisits a particular GUI might not need similar coaching as to what one has received during the first visit. This is particularly true when the learner revisits the GUI just to confirm what one has selected or typed in is correct. To handle this challenge gracefully, the agent's interventions are “localized” to each GUI aiming at capitalizing learners’ learning experiences. Thus, the DDN depicted in Figure 7 has decision nodes ($D^n_t$) and utility nodes ($U^n_t$) resides in each time-slice. The arcs directed from $H^U_t, \zeta^U_t, \zeta^U_t$ and $K^U_t$ to $H^U_t$, capturing the idea that the agent’s satisfaction depends on the acquisition level of $H$ and $\zeta$ and subsequently tailored interventions can be provided at a particular GUI. The configuration encoded in the utility nodes of the DNs in INQPRO will result in the agent’s preference to be “Doing Nothing” whenever there is no evidence instantiation or during a learner’s $H$ and $\zeta$ are assessed to be “mastery”. Such encoding methodology follows exploratory learning approach where learners are given the freedom to explore, and subsequently make more meaningful learning while interacting with INQPRO. Conversely, if a learner’s $H$ and $\zeta$ are assessed to be “non-mastery”, indirect hint will be prompted accordingly.

Generating DDN during Runtime

Figure 8 depicts the computer algorithm for generating a DDN during runtime. The CreateDDN function is called each time a GUI is accessed, which in this context refers to the event when a learner clicks at the “Next” or “Go To…” button. As depicted in Figure 8, the CreateDDN function takes into two parameters: $P$ and $t$. $P$ denotes the next GUI to be displayed while $t$ denotes the n$th$ time $P$ is accessed.

The CreateDDN function firstly checks whether or not the runtime DDN ($P_{DDN}$) exists (refer to line 8, Figure 8). If it does not, a copy of Scenario DN will be duplicated for $P_{DDN}$. Having set Scenario DN as the first network in $P_{DDN}$ indicates that it is the first time the learner interacts with INQPRO.

If $P_{DDN}$ is found, the decision nodes ($A_{DDN}$) and utility nodes ($Y_{DDN}$) in $P_{DDN}$ will firstly be removed. Removing these nodes greatly reduces the computational time in updating the DDN. This is particularly crucial as lagging in updating $P_{DDN}$ will affect the smooth interaction process between learners and INQPRO. The AppendNodes function will then update $P_{DDN}$ with nodes in DN which corresponds to $P$. The newly added nodes can be differentiated from similar nodes in previous time-slices by the parameter $t$, which specifies the nth $P$ is accessed. Once the nodes in $P$ are appended to $P_{DDN}$, the DirectArc function create causal arcs directed from the static nodes ($S^H, S^F, and S^V$) to the dynamic nodes ($H^F, H^U, and H^F$) in $P$. The LoadCPT function defines the Conditional Probability Tables (CPTs) for the dynamic nodes in the current GUI. At this stage of our research work, the Conditional Probability Tables (CPTs) are defined by domain experts and vary from one GUI to another. This indicates that the value of the dynamic node at one time can only affect the value of the similar dynamic node at the next time step depending on the GUIs accessed.
Algorithm CreateDDN \((P, t)\)

\[\text{Input: } P \in \{\text{GUIs except the Feedback GUI}\}\]

\[t = \text{the } n\text{th time } P \text{ is accessed}\]

\[P_{DDN} \leftarrow \text{DDN generated during runtime}\]

\[\Gamma_{DDN} \leftarrow \text{array of GUIs } \{\Gamma_1, \ldots, \Gamma_n\} \text{ except } P\]

\[\Delta_{DDN} \leftarrow \text{array of decision nodes } \{\Delta_1, \ldots, \Delta_n\} \text{ in } \Gamma\]

\[\Upsilon_{DDN} \leftarrow \text{array of utility nodes } \{\Upsilon_1, \ldots, \Upsilon_n\} \text{ in } \Gamma\]

If Not Exist \(P_{DDN}\) then

\[P_{DDN} \leftarrow \text{AppendNodes} \left( P_{DDN}, Sce \right)\]

Else

\[\delta \leftarrow \text{RemoveDecisionNodes} \left( \Delta \right)\]

\[\text{RemoveUtilityNodes} \left( \Upsilon \right)\]

\[P_{DDN} \leftarrow \text{AppendNodes} \left( P_{DDN}, \{P, t\} \right)\]

\[\text{DirectArc} \left( \{S^R \rightarrow H^F\}, \{S^F \rightarrow P^F\}, \{S^R \rightarrow H^F\} \right)\]

\[\text{LoadCPT} \left( P, t \right)\]

End if

\[\text{Output: } P_{DDN} \text{ with } P \text{ added.}\]

Figure 8. Algorithm for generating runtime DDN

Although the above mentioned algorithm is tailored for INQPRO, it is applicable and transferable to any learning environment consisting of the following three properties:

(i) There is a set of GUIs \(G\), such that each GUI \(g \in G\) is distinguishable. That is, the learning environment consists of several GUIs and all the GUIs having different interface design.

(ii) There is a set of DNs \(D\), such that each DN \(d \in D\) is distinguishable, and \(g\) has a corresponding \(d\). Each \(d\) is unique.

(iii) There is an identical set of random variables that evolve across time \(\{H_1, \ldots, H_n\}\) for every \(d\).

(iv) A DDN is made up of interconnected time-slices of static DNs, where \(\{H_1, \ldots, H_n\}\) in every two consecutive time-slices are linked together through temporal arcs. Introduce a set of static nodes \(\{S_1, \ldots, S_n\}\) so that arcs can be directed from the static nodes to the corresponding \(\{H_1, \ldots, H_n\}\). However, the dependencies between \(\{S_1, \ldots, S_n\}\) are domain dependent.

In the next section, we shall present a walk-through with INQPRO in order to provide reader with a better understanding of how the proposed DDN model behaves, which in this context, predicting the learner’s mastery level of scientific inquiry skills and provision of appropriate support in timely manner.

A Sample Walk-Through with INQPRO

The following scenario \(P\) aims at illustrating how the proposed DDN model can be employed to assess temporally variable \(H, \varsigma\) and \(K\). To simplify the explanation, we shall scope our illustration to only five GUIs accessed by both learners \(\Lambda_A\) and \(\Lambda_B\) (Figure 9). Let us presume that \(\Lambda_A\) denotes a learner whose mastery levels of \(H, \varsigma\) and \(K\) have increased while \(\Lambda_B\) represents a learner who has failed to achieve meaningful learning after interacting with INQPRO.

Scenario \(P\): Supposing that two learners \(\Lambda_A\) and \(\Lambda_B\) have interacted with INQPRO and both demonstrate identical learning path (Figure 9). In Figure 9, both \(\Lambda_A\) and \(\Lambda_B\) have started with the Scenario (Sce) GUI and later proceeded to Hypothesis Visualization (Vz), Formula (Fe), Scenario (Sce), and finally the Data Comparison (Dc) GUI. Presumably in time-slice \(t_i\), both \(\Lambda_A\) and \(\Lambda_B\) would have constructed inappropriate hypothesis statements, and unable
to identify correct variables types. However, having performed a series of meaningful interactions (e.g. revisiting the previously formulated hypothesis statement, selecting appropriate variable types, and interacted to the Agent), Λₐ has ultimately reconstructed an appropriate hypothesis statement during her revisit to Scenario GUI in time-slice \( t₄ \). Conversely Λₐ who did not demonstrate meaningful interactions and has ignored hints generated by the Agent, has failed to reconstruct an appropriate hypothesis statement in time-slice \( t₄ \).

Figure 9. The INQPRO GUIs accessed by both learners Λₐ and Λₐ. Assuming that after interacting with INQPRO, Λₐ is classified as “mastery” while Λₐ is classified as “non-mastery”.

Figure 10. The DDN generated after accessing five GUIs. Both DN structures and temporal nodes are different in all the time-slices but are presented in the same structure for ease of presentation.

Figure 10 depicts the run-time generated DDN model that aims at modeling a learner’s Η, ζ, and Κ. In this model, there are a total of 15 dynamic nodes (e.g. nodes \( H_{S}^{t₁}, H_{S}^{t₂}, H_{S}^{t₃}, H_{S}^{t₄}, H_{S}^{t₅} \) and 3 Static nodes (nodes \( S^H, S^ζ, S^K \)). The Static nodes capture the gradual change in the mastery levels of Η, ζ, and Κ for the five time-slices through the “accumulative” modeling of the scientific inquiry skills. In the above scenario, given that in time-slice \( t₁ \) that Λₐ and Λₐ have performed unsuccessful interactions, which include the inappropriateness of constructing hypothesis statement, selecting variable types, and answering questions prompted by the Agent incorrectly, a non-mastery level is suggested by the DDN for both Η (node \( H_{S}^{t₁} \)) and ζ (node \( ζ_{S}^{t₁} \)). Once the beliefs of the DDN are updated, knowing the posterior probabilities of \( H_{S}^{t₁} \) and \( ζ_{S}^{t₁} \) would allow the posterior probabilities of \( S^H \) and \( S^ζ \) to be calculated and subsequently the ultimate mastery levels of Η and ζ are reviewed. This is possible due to the conditional effects of the Static nodes (nodes \( S^H \) and \( S^ζ \)) on the dynamic nodes (nodes \( H_{S}^{t₁} \) and \( ζ_{S}^{t₁} \)) as depicted by the arcs directed from the Static nodes to dynamic nodes.

Apart from knowing the mastery levels of Η and ζ, updating the beliefs in time-slice \( t₁ \) has also provided a unique mechanism for the Agent to select the decision alternative (node \( E_{S}^{t₁} \)) that maximizes the expected utility (node \( E_{U}^{t₁} \)). In the above scenario, the Agent has suggested both Λₐ and Λₐ to restudy the hypothesis formulated and variables selected in time-slice \( t₁ \) due to the low mastery levels.

Presumably in the Hypothesis Visualization GUI, both Λₐ and Λₐ have equally interacted with the computer simulations by performing the drag-and-drop event on the different masses, identified different graph patterns generated as a result of different masses, and correctly answered questions prompted by Agent. Once such information is interpreted into the DDN, the mastery levels of Η, ζ, and Κ in time-slice \( t₂ \) for both Λₐ and Λₐ can be
Due to identical interactions elicited by both learners, the mastery levels and probability values are similar in time-slice $t_2$.

![Figure 11](image)

Figure 11. Probability assessment of $H$, $\varsigma$, and $K$ for (a) $\Lambda_A$ (b) $\Lambda_B$. The dotted lines indicate the threshold value (33.33%) for $H$, $\varsigma$, and $K$ to be categorized as “mastery”.

After studying the computer simulations, the learners proceeded to the Formula Investigation GUI. The interaction logs capture the information acquired when $\Lambda_A$ explores the relationships between $m$, $k$ and $T$, and react positively to the questions prompted by the Agent. Conversely, $\Lambda_B$ fails to define the relationships between the variables given by the formula and wrongly answers the questions prompted by the agent. By interpreting this evidence into the DDN, the INQPRO suggested a “partial mastery” level for $\Lambda_A$ while $\Lambda_B$ remains “non-mastery” for $H$, $\varsigma$, and $K$.

Presumably that in time-slice $t_4$, INQPRO has captured meaningful interactions demonstrated by $\Lambda_A$ when $\Lambda_A$ revisits the Scenario GUI. These interactions include modifying the previously formulated hypothesis statement to a correct one and appropriately identifying variables. By taking the previous learning experiences (time-slice $t_3$) and current evidence into consideration, INQPRO has estimated a 45% mastery level for $\Lambda_A$, 55% for $\varsigma$, and 48% of $K$ for $\Lambda_A$. $\Lambda_B$ who has also revisited the Scenario GUI conversely, did not manage to formulate the correct hypothesis and as such has resulted in the “non-mastery” level for $H$.

We shall present the evaluation of the proposed DDN model in the next section. A two-phase evaluation process was conducted aiming at investigating the matching accuracies and consistency of the proposed model in classifying learners’ $H$, $\varsigma$, $K$, and $A$. We computed the matching accuracies based on a total of 76 learners and 6 domain experts.

**Evaluation of the Proposed DDN Model**

The evaluation phase of this research work aims at investigating the performance of our proposed DDN model for the INQPRO learning environment. At the first phase of evaluation $\Pi_1$, 76 first-year university learners participated in the evaluation while in the second phase of evaluation $\Pi_2$, there were a total of 31 first-year university learners involved. In both evaluation phases, learners were categorized into “High”, “Moderate”, and “Low” academic performance based on their O’level science or physics results. For both evaluation phases, learners participated in sessions that lasted between 80 to 100 minutes consisting of an introduction session to INQPRO, a pretest, a session with INQPRO and a posttest. Both the pretest and posttest consisted of 23 multiple choices questions with 9 questions targeted at assessing $H$, while 14 questions for $\varsigma$. Due to the dependency between $H$, $\varsigma$, and $K$, knowing the mastery levels of $H$ and $\varsigma$ will help in predicting the mastery level of $K$. In this study, the pretest and posttest were validated by 2 domain experts who have years of experience in teaching Physics and Science. Learners who have scored higher than 75% for $H$, $\varsigma$, and $K$ in both tests will be graded as “mastery” whereas those who scored less than 45% were regarded as “non-mastery”. Before interacting with INQPRO, learners were requested to elicit their own
mastery levels of Κ, Η and ζ (hereafter ‘Pre-INQPRO self-ranking’). After the session with INQPRO, again, the learners were asked to rank their own acquisition level of Κ, Η and ζ (hereafter ‘Post-INQPRO self-ranking’). A 3-rank scale (mastery, partial mastery, non-mastery) was given to the learners to assist them during the self-ranking process. In this study, however, the learners’ self ranking of mastery levels for Κ, Η and ζ was not a primary index to calculate the performance of the proposed DDN model due to the fact that there were implicit and explicit factors which influenced the elicitation process.

In addition to the field tests administrated to the learners in Π1 and Π2, we consulted six domain experts to elicit their expectations on the proposed DDN model. At Π1, the experts were presented with a 32 randomly selected learner interaction logs and videos, while at Π2, there were a total of 14 randomly selected interaction logs were given to experts. The entire process lasted between 75 to 110 minutes.

At each phase of evaluation, we ran two experiments to obtain a more precise picture of the overall performance and stability of the proposed DDN model in assessing Η, ζ, and Κ of different categories of learners.

The first experiment aimed at examining how accurate and sensitive the proposed DDN model in classifying Η, ζ, and Κ for different categories of learners. To achieve this aim, we calculated the model’s matching accuracies by taking results obtained from the pretest and posttest of 107 learners as a reference. In this experiment, we computed the first set of the model’s matching accuracies by comparing the classifications given by the model with the classifications obtained from the pretest. In detail, those states with the highest probabilities for the Static nodes (S_A, S_B, and S_C) Figure 7) were compared to the classifications given by the pretest. At this stage of the experiment when the comparison is carried out, the DDN consisted of only nodes from the Scenario GUI mainly because the Scenario was the first GUI accessed by learners upon logging into the INQPRO learning environment. With the assumption that the actual learning has yet to occur for a particular learner within the Scenario GUI, the mastery levels of Η, ζ, and Κ should be reflected by the Static nodes through the states with the highest probabilities. Hence, we argue that such comparison approach is considered valid. Similarly, after interaction with INQPRO, the states of the Static nodes with the highest probabilities denote the ultimate mastery level of Η, ζ, and Κ for a particular learner. Again, we compare the classifications given by the model with the results obtained from the posttest to obtain the second set of the model’s matching accuracies. Having these two sets of matching accuracies, the overall average performance of the DDN model was computed and presented in terms of accuracies for Η, ζ, and Κ.

The second experiment aimed at obtaining the matching accuracies for Η, ζ, and Κ from the experts’ point of view. In addition, this experiment also helped in identifying the appropriateness and suitability of decision alternatives (node S^A_B), which in this context were the pedagogical interventions A generated by the Agent. The domain experts were consulted to elicit their predictions on the mastery levels of Η, ζ, Κ, and A at the GUI level. By studying the interaction logs for each GUI navigated by a learner, the experts verified the appropriateness of the pedagogical interventions (A) generated and estimated the mastery levels of Κ, Η and ζ. The matching accuracies of Κ, Η, ζ, and A were then calculated by counting the classifications given by a particular DDN model that matched those elicited by the experts. In addition to that, graphs (example see Figure 11 (a) and (b)) illustrating the evolving mastery levels of Κ, Η and ζ were given to the experts aiming at investigating whether or not the patterns depicted fit their expectations. For each GUI, a√was assigned to the matched classifications while an x was assigned to those misclassified.

Results

In this section, we shall firstly discuss the learners’ overall performance before and after interacting with INQPRO. Statistical significance tests were performed at both phase of evaluation to compare the individually matched improvements of the learners from pretest to posttest. Because the same pair of learners completed both a pretest and a posttest, a one-tailed paired difference experiment was performed to gauge the significant of the improvement.

We then present and analyze the accuracies of the proposed DDN model in classifying the mastery levels of Κ, Η and ζ for different categories of learners. We shall begin with the first phase of evaluation Π1 and subsequently with the second phase of evaluation Π2. The results of the experiments at each evaluation phase shall be addressed and
discussed in detail. In this study, we use the pretest and posttest results as benchmark for the computation of matching accuracies. The self ranking, however, is not a benchmark to how accurate the proposed DDN performs due to the learners’ implicit and explicit factors that might influence the learners in self ranking process.

First Phase of Evaluation ($\Pi_1$)

Figure 12 depicts an overview performance demonstrated by the learners at the first phase of evaluation. The mean score for pretest was 16.3 while the mean score for posttest was 36.4. The standard deviation for pretest was relatively high (10.9) compare to the posttest with 6.7. It was found that the $p$-value was $1.2\times10^{-7}$ and thus concluded that the t-test confirmed that the improvement from pretest to posttest was significant at a $p<0.05$ level.

![Figure 12. Mean pretest and posttest scores for 46 learners](image)

Experiment 1: Model’s matching accuracies using pretest and posttest as reference

Table 1 shows the results of the proposed DDN model in classifying different categories of learners. As shown in Table 1, the model demonstrated an overall matching accuracies of higher than 50% for the pretest, posttest, pre-INQPRO self ranking, and post-INQPRO self ranking. The high matching accuracies given at the pretest suggested that the proposed DDN performed well in classifying learners’ initial acquisition of $\Phi$ (97.8%) and the understanding of overall scientific inquiry skills $\mathbf{K}$ (93.5%). The variation of matching accuracies for $\mathbf{K}$ and $\Phi$ within 10% with respect to the self-ranking columns has also suggested the model’s consistency in modeling learners’ scientific inquiry skills. In this experiment, the reliability of self ranking scores is not accountable due to personal traits. Thus, taking them away would give the overall matching accuracy for $\mathbf{K}=85.9\%$ while $\Phi=86.9\%$. Such promising results indicate that the proposed DDN model was able to perform well in classifying learners’ $\Phi$ and $\mathbf{K}$.

Table 1 has, however, depicted a relatively low matching accuracy for $\varsigma$ with respect to the pretest (52.2%). Such low accuracy was mainly due to the misclassification of learners into “partial-mastery” by the DDN model whereas the pretest categorized them as “non-mastery”. Through the interview sessions with randomly selected learners, we found out that the learners learned from pretest on $\Phi$ and $\varsigma$ and such learning effect had subsequently become the prior knowledge as they interacted with INQPRO. As a result, misclassification happened and a low accuracy for $\varsigma$ at the pretest is shown (Table 1, $\varsigma=52.2\%$). Such phenomenon was also reflected by the low accuracy depicted at the pre-INQPRO self ranking for $\varsigma$ (47.8%). These learners had ranked themselves as “non-mastery” before the pretest.

The results shown in Table 1 also highlights that the proposed DDN model performed equivalently well in classifying learners from the moderate and low category. Taking the moderate group as an example, the proposed
DDN model successfully classified 91.7% for K, 83.0% for H, and 83.3% for \(\varsigma\) with respect to the Posttest column. The ability to classify learners from different categories is a crucial part for a good classifier.

Table 1. The model’s matching accuracies with respect to results obtained from pretest, posttest, and self-ranking during the first phase of evaluation PI (46 learners)

<table>
<thead>
<tr>
<th>Scientific Inquiry Skill</th>
<th>Category of learners</th>
<th>Pretest</th>
<th># and % of correct classification based on</th>
<th>Posttest</th>
<th>Pre-INQPRO</th>
<th>Post-INQPRO</th>
<th>Self-Ranking</th>
<th>Post-INQPRO</th>
<th>Self-Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>Advance (n=13)</td>
<td>10(90.9)</td>
<td>98(81.8)</td>
<td>2(18.2)</td>
<td>7(36.6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Moderate (n=20)</td>
<td>11(91.1)</td>
<td>11(91.1)</td>
<td>7(58.3)</td>
<td>7(58.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Low (n=43)</td>
<td>22(95.7)</td>
<td>16(69.6)</td>
<td>15(65.2)</td>
<td>9(39.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>43(93.5)</td>
<td>36(78.3)</td>
<td>24(52.2)</td>
<td>23(50.0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Advance (n=13)</td>
<td>11(100.0)</td>
<td>9(82.0)</td>
<td>2(18.2)</td>
<td>7(36.6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Moderate (n=20)</td>
<td>11(91.1)</td>
<td>10(83.0)</td>
<td>7(58.3)</td>
<td>6(50.0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Low (n=43)</td>
<td>23(100.0)</td>
<td>16(69.6)</td>
<td>14(60.9)</td>
<td>14(60.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>45(97.8)</td>
<td>35(76.1)</td>
<td>23(50.0)</td>
<td>27(58.7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\varsigma)</td>
<td>Advance (n=13)</td>
<td>8(82.7)</td>
<td>10(90.9)</td>
<td>7(63.6)</td>
<td>9(81.8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Moderate (n=20)</td>
<td>5(41.7)</td>
<td>10(83.3)</td>
<td>5(41.7)</td>
<td>6(50.0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Low (n=43)</td>
<td>11(47.8)</td>
<td>11(47.8)</td>
<td>10(43.5)</td>
<td>18(78.3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>24(52.2)</td>
<td>31(67.4)</td>
<td>22(47.8)</td>
<td>33(71.7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As a conclusion, the proposed DDN model demonstrated its consistency in classifying different categories of learners with an average accuracy of 85.9% for K, 86.9% for H while 59.8% for \(\varsigma\). In the next subsection, we shall address the second evaluation objective by investigating the behaviours and matching accuracies of the proposed DDN model at GUI level elicited from the domain experts. This step is crucial as it helps in verifying the consistency and appropriateness of the proposed DDN model before conclusion on the performance is made.

Experiment 2: Model’s matching accuracies through domain experts

The importance of having domain experts participating in this study is two-fold: (i) to verify the classifications depicted by the model at each individual GUI given a set of learner interactions as evidences; (ii) to examine the appropriateness and suitability of pedagogical interventions generated by the Agent at each GUI. Such elicitation process has allowed us to study how similar the pedagogical interventions generated by the model as compared to those from the experts.

Table 3. The model’s matching accuracies with respect to domain experts

<table>
<thead>
<tr>
<th>Advance (n=4)</th>
<th>Moderate (n=12)</th>
<th>Weak (n=2)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>91.7</td>
<td>84.4</td>
<td>91.7</td>
</tr>
<tr>
<td>H</td>
<td>97.2</td>
<td>80.5</td>
<td>95.8</td>
</tr>
<tr>
<td>(\varsigma)</td>
<td>91.3</td>
<td>80.3</td>
<td>87.5</td>
</tr>
<tr>
<td>A</td>
<td>98.4</td>
<td>93.4</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3 shows the matching accuracies of the proposed DDN model as elicited by the domain experts. The model has depicted equivalent promising results for the three categories of learners with respect to K (89.3%), H (91.2%), \(\varsigma\) (86.4%), and A (97.3%). With the matching accuracies exceeding 80%, we can ascertain that the proposed DDN model had demonstrated its adaptability in classifying K, H, and \(\varsigma\) that evolved across time. In addition, the high matching accuracy depicted for A (96.1%) in Table 3 also indicated that the experts had somehow agreed with the pedagogical interventions generated by the Agent. During the interview sessions, the experts had commented that it was a difficult process to justify whether or not a particular learner had achieved mastery levels of \(\varsigma\) and H until the
correct hypothesis statement was constructed and appropriate variables were selected. Due to the constraint, low matching accuracies were recorded for H (80.5%) and ς (80.3%) for the moderate group. When the domain experts were asked regarding the effects of interacting with the Hypothesis Visualization GUI to the acquisition of K, H, and ς, they strongly agree that active interactions could help improve learner’s acquisition of scientific inquiry skills. However, again, the domain experts require solid evidence before concluding the mastery level of a learner.

**Evaluation Phase Two (Π2)**

Figure 13 depicts an overview performance demonstrated by the learners at the second phase of evaluation. The mean scores for both pretest and posttest at the second phase of evaluation were relatively low compared to the first phase of evaluation. However for both evaluation phases, the standard deviations were similar for both pretest and posttest. When t-test was performed, it was found that the p-value was 7.8E-05 and thus concluded that the overall improvement demonstrated by the learners from pretest to posttest was significant at a p<0.05 level.

![Figure 13. Mean pretest and posttest scores for 31 learners](image)

**Experiment 1: Model’s matching accuracies using pretest and posttest as reference**

A total of 31 learners participated at Π2. Again, the matching accuracies of the model depicted in Table 4 were calculated by comparing the classifications given by the model with those obtained from pretest, posttest, and self-ranking scores.

Table 4 shows the matching accuracies of three different categories of learners. Without taking the matching accuracies of self-ranking into consideration, the average matching accuracies for both H and ς reached 85.5%. As shown in Table 4, the overall matching accuracies for ς was again relatively low (pretest=64.5%; pretest self ranking = 58.1%) as compared to the accuracies depicted by H and K. The reason is identical to that we have analyzed and concluded in Π1 where the “learning” effect experienced by the learners during the pretest had somehow helped them recall scientific inquiry skills. When such “learning” effect was transferred and incorporated into the DDN model, the mismatch of classifications occur.

Another point worth highlighting in this evaluation is the relatively low matching accuracy for H (74.2%) shown in Table 1. There were two major reasons for this issue. Firstly, from the observation as the learners worked individually with INQPRO, we found out that there were learners who did not perform meaningful interactions. These learners did not interact with the Agent, and had not performed enough interactions such as drag-and-drop the different masses at the Hypothesis Visualization GUI. Due to the lack of interactions, INQPRO would treat the learners as “unable to interact successfully” and when these evidences are fed into the network, the resulting
categories were mostly likely to be either “partial mastery” or “non mastery”. The second reason responsible for the low accuracy was due to the uncompleted posttest. At this evaluation phase Π2, there were two learners who did not complete the posttest. As a result, the mismatch of classifications occurred. However, if these two reasons were not taken into consideration when it came to calculating the matching accuracies, the matching accuracy for H for the posttest section would then be 88.5%.

Table 4. The model’s matching accuracies with respect to the results obtained from pre-test, post-test, and self-ranking during the second phase of evaluation Π2 (31 learners)

<table>
<thead>
<tr>
<th>Scientific Inquiry Skill</th>
<th>Category of learners</th>
<th># and % of correct classification based on Pretest</th>
<th>Posttest</th>
<th>Pretest Self-Ranking</th>
<th>Posttest Self-Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>Advance(n=8)</td>
<td>7(87.5)</td>
<td>7(87.8)</td>
<td>5(62.5)</td>
<td>4(50.0)</td>
</tr>
<tr>
<td></td>
<td>Moderate(n=14)</td>
<td>11(78.6)</td>
<td>13(92.9)</td>
<td>7(50.0)</td>
<td>7(50.0)</td>
</tr>
<tr>
<td></td>
<td>Low(n=9)</td>
<td>8(88.9)</td>
<td>7(77.8)</td>
<td>7(77.8)</td>
<td>4(44.4)</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>26(83.9)</td>
<td>27(87.1)</td>
<td>19(61.3)</td>
<td>15(48.4)</td>
</tr>
<tr>
<td>H</td>
<td>Advance(n=8)</td>
<td>7(87.5)</td>
<td>6(75.0)</td>
<td>6(75.0)</td>
<td>5(62.5)</td>
</tr>
<tr>
<td></td>
<td>Moderate(n=14)</td>
<td>14(100.0)</td>
<td>12(86.0)</td>
<td>5(35.7)</td>
<td>7(50.0)</td>
</tr>
<tr>
<td></td>
<td>Low(n=9)</td>
<td>9(100.0)</td>
<td>5(55.6)</td>
<td>6(66.7)</td>
<td>5(55.6)</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>30(96.8)</td>
<td>23(74.2)</td>
<td>17(54.8)</td>
<td>17(54.8)</td>
</tr>
<tr>
<td>ζ</td>
<td>Advance(n=8)</td>
<td>5(62.5)</td>
<td>7(87.5)</td>
<td>3(37.5)</td>
<td>8(100.0)</td>
</tr>
<tr>
<td></td>
<td>Moderate(n=14)</td>
<td>8(57.1)</td>
<td>13(92.9)</td>
<td>8(57.1)</td>
<td>13(92.9)</td>
</tr>
<tr>
<td></td>
<td>Low(n=9)</td>
<td>7(77.8)</td>
<td>5(55.6)</td>
<td>7(77.8)</td>
<td>8(88.9)</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>20(64.5)</td>
<td>25(80.7)</td>
<td>18(58.1)</td>
<td>29(93.6)</td>
</tr>
</tbody>
</table>

Experiment 2: Model’s matching accuracies through domain experts

Table 5 shows the averages for the matching accuracies for K (93.3%), H (83.1%), ζ (82.9%), and A (85.4%). With all the average accuracies exceeding 80%, once again, the proposed DDN model was highly rated for its appropriateness in modeling learners’ scientific inquiry skills and adaptability in providing pedagogical interventions for different categories of learners. In this experiment, again, the experts rated ζ (82.9%) and H (83.1%) with a relatively low accuracy due to the similar reasons given at Π1. Compared to Π1, Π2 has more learners who did not navigate back to the Scenario GUI to correct the hypothesis statements. In addition, with only one learner categorized under the weak category, this again had contributed to the relatively low accuracy recorded for H and ζ (Table 5).

Table 5. The model’s matching accuracies with respect to domain experts

<table>
<thead>
<tr>
<th></th>
<th>Advance (n=4)</th>
<th>Moderate (n=9)</th>
<th>Weak (n=1)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>93.2</td>
<td>95.7</td>
<td>90.9</td>
<td>93.3</td>
</tr>
<tr>
<td>H</td>
<td>85.5</td>
<td>86.4</td>
<td>77.3</td>
<td>83.1</td>
</tr>
<tr>
<td>ζ</td>
<td>84.9</td>
<td>91.0</td>
<td>72.7</td>
<td>82.9</td>
</tr>
<tr>
<td>A</td>
<td>92.4</td>
<td>90.5</td>
<td>73.3</td>
<td>85.4</td>
</tr>
</tbody>
</table>

Conclusion and Future Directions

The research work presented in this article aimed at proposing a methodological approach for modeling and intervening under uncertainty within the INQPRO learning environment. In this study, we scope the scientific inquiry skills to Hypothesis Formulation H and Variables Identification ζ. During the model construction phase, we introduced K to represent the overall understanding of H and ζ because of the causal relationship between them. In this article, we highlighted how our proposed methodological approach has complemented and contributed to the
existing research work in science education and intelligent tutoring systems particularly on those leveraging decision-theoretic approach. From science education perspective, assessing a learner’s scientific inquiry skills had always been a challenging task (de Jong, 2006). The pencil-and-paper approach is often deployed as an assessment method. Such method, although commonly implemented, has its limitations as it only allows the assessment of the final but not the evolving mastery levels of scientific inquiry skills. In this light, one of the aims of the proposed methodological approach was to provide a mechanism to explicitly visualize the evolving mastery level as the learners interact with a particular computer based learning environment. From the ITS perspective, most of the existing decision-theoretic tutoring systems rely on a DDN or DBN that is constructed from a repeated static BN or DN. This approach, however, is not always practically sound to all the learning environments. The INQPRO deployed in this study for instance, consists of six different GUIs with different DNs correspond to them.

Overcoming these challenges is not a trivial task. Various considerations in constructing the DDN need to be well taken care of. In this study, the proposed novel methodological approach consisted of (i) a dynamically generates a DDN from individual DNs at run-time basis. Once a particular GUI was navigated by a learner, the corresponding DN is appended to the newly created DDN; (ii) inclusion of the static nodes in the newly created DDN at run-time basis. Once the DDN was generated, the static nodes were added on top of the nodes aggregated from separated DN.

In this article, we have presented two evaluation phases, Π1 and Π2, that aimed at investigating the behaviours and performance of the proposed DDN model. A total of 46 learners participated at Π1 while 31 learners participated at Π2. There were a total of 6 domain experts participating at both evaluation phases. At each evaluation phase, we conducted 2 experiments. The first experiment relied on the pretest, posttest, and self ranking scores as a benchmark for calculating the matching accuracies while the second experiment depended on the classification results given by the domain experts as a reference. For both evaluation phases, we found out that the matching accuracy for ζ with respect to the pretest was relatively low. This was mainly due to the misclassification of learners into “partial-mastery” by the DDN model whereas the pretest categorized them as “non-mastery”. The interview sessions with randomly selected learners from both phases revealed that the learners learned about ζ and H from the pretest, and such learning effect became the prior knowledge that equipped them with the necessity to interact with INQPRO. A slightly low matching accuracy for ζ and H at the posttest for both evaluation phases is mainly because of the unexpected system faults such as the runtime errors, and program termination. Apart from that, the relatively low matching accuracy for H at the posttest was also due to the incompleteness of meaningful interactions with INQPRO demonstrated by the learners. Due to lack of evidence, the system interpreted these learners as “non-mastery”. These learners, although they had not interacted successfully with INQPRO, had scored well at the posttest. Due to the mismatched of such circumstances, the accuracy of H at the posttest was relatively low.

Employing DDN approach in this study, however, created a contradiction to the conventional assessment method employed by domain experts. Employing DDN in this study has allowed the modeling of Π2, and K, and ζ across time even without the existence of primary evidence. That is, although the learners did not specifically navigate back to the Scenario GUI to rephrase the hypothesis statement and select the different types of variables, the model has high confidence in predicting the mastery levels of the scientific inquiry skills with only the existence of alternative evidence such as the drag-and-drop, mouse-clicking, and mouse-move events.

The current research work can be further enhanced and tackled from two perspectives: (i) the structural aspect of the proposed DDN model; and (ii) the parameters or beliefs of the nodes in each DN. The current DDN model is having d-separation connection for the static nodes (H → K, and ζ → K). The arc stretching from ζ to H indicates a causal relationship between these two nodes, which suggests that increasing the probability of ζ will also increase the probability of H. The proposed approach has, however, resulted in the tendency of the proposed model to over interpret of evidence upon H due to the propagation of probabilities from ζ to H and from K to H.

One of the great advantages of employing the proposed approach is that it can be directly deployed without the need to train the model beforehand. In this study, the parameters setting of the proposed DDN model can be incorporated from the knowledge elicited from the domain experts. However, such a method does not guarantee an optimal solution to model and intervene under uncertainty within the INQPRO learning environment. Thus, our immediate next step is to apply the Bayesian machine learning approach. We hope that by applying the machine learning approach to learn both the structure and the beliefs of the DNs, the human interventions during the model
construction can be minimized and at the same time obtaining an optimum solution to the modeling and intervening issues can be obtained.

References


