Bayesian Network Student Modelling on Intelligent Tutoring System

Heri Wijayanto[1](✉), I G. P. Suta Wijaya[1], Siti Nurmutmainnah[2], Ida Bagus Ketut Widiartha[3], and Ramaditia Dwiyansaputra[1]

[1] Department of Informatics Engineering Faculty of Engineering Mataram, Mataram, Indonesia
heri@unram.ac.id
[2] Department of Electrical Engineering Faculty of Engineering Mataram, Mataram, Indonesia
[3] Department of TEMEP, College of EngineeringSeoul National University Seoul, Seoul, South Korea

Abstract. Intelligent Tutoring System (ITS) is an electronic tutorial that has intelligence for adapting learning materials. It adjusts the contents of learning materials depending on the users' requirements. This research developed an ITS for Object Oriented Programming course that is based on Bayesian Network as its inference engine. It has run well which is shown by the variation of the study tracks of each student based on the Bayesian Network Student Model. This tutorial is also divided into two part, there are basic parts that contains basic programming technique and the advanced part that provides the concepts of Object Oriented Programming. Based on those two parts, this study wants to examine their differences of them. By the Levenshtein distance, the highest distance between students' ways of learning is 11 and its average is 5.385 for the basic part of Programming and 4.8 for the advanced part of the advanced part. The advanced part also has a positive value of skewness of the frequency distribution, it is 0,81 (left skew), which means that this distance is majority short. Whereas the basic of programming has a negative value that is −0.68 (right skew), in the other words, it shows that the elementary part has a longer distance of learning way. Data of the distances of advanced material parts concentrated on the second quartile by median is 4, whereas the basic part, by median is 6, data is concentrated in the third quartile. Mean Opinion Score shows that students are more interested to use the ITS than the classical Tutoring System, it is because the ITS has an average value is 8 (very agree) and the classical tutorial is 7 (agree). Besides that, the other criteria such as user-friendly, speed, report response, and appearance have 8 scores or very agree, and only a criterion has 7 (agree) in the stability.

Keywords: Intelligent Tutoring System · Bayesian Network · Object Oriented Programming

I G. P. Suta Wijaya et al. (Eds.): MIMSE-I-C 2022, ACSR 102, pp. 298–310, 2022.
https://doi.org/10.2991/978-94-6463-084-8_26

1 Introduction

Artificial intelligence is a branch of study in computer science to make a computer able to act like a human to solve problems. In this research, artificial intelligence is applied to an educational tool that is called the ITS (Intelligent Tutoring System), so this application is hoped to give private learning to students. In the education field, especially for developing countries that have problems with the lack of infrastructures and instructors, this application is expected to give a solution.

The disparity between the ITS and the other tutoring systems on the internet is the capability in applying the pedagogic substance. The customary tutorial just conducts a lesson tutoring in a series without feedback that adjusts the student's study path by the student achievement. On the other hand, intelligence tutoring can adapt the materials to give the appropriate learning for each student individually. It is because a student is modeled by this kind of tutorial that uses some methods. Therefore, the lessons that are given are different for each student [1].

Object Oriented Programming (OOP) is an obligatory course for informatics students. The progress of programming language is much related to this course because the problem that programmers face becomes more complicated. Besides that, the technology of hardware development supports the OOP method that needs high-level hardware specifications.

Teaching OOP is not simple because the lecturer must change the mindset of students from structured programming to the object paradigm. In OOP, the programmer sees the program of the computer as an association of objects that are related to other objects.

This research examines two attributes of ITS, which are the variety of learning paths and the student opinions about ITS. The variety of learning paths is measured by the Levenshtein distance and the Mean Opinion Score (MOS) is used to know student opinions about ITS. By knowing the distance of learning paths among students, it is hoped that this research result can be used by teachers or other systems for clustering student learning styles so that the systems or human teachers can give appropriate advanced learning methods.

2 Related Works

2.1 Intelligent Tutoring System

An Intelligent Tutoring System (ITS) is the next generation of Computer Aided Instruction (CAI) [2]. The main feature of ITS is its capability to adapt to user learning styles so that it can give the appropriate learning for students.

Adaptation of an ITS is needed because each student has a unique characteristic, for example, the learning style has been examined by Garcia [3]. Garcia used Bayesian Network (BN) for diagnosing student learning styles so that the ITS can give the appropriate learning method. Garcia's method for detecting learning style refers to Felder's learning style. Besides that, Patti West uses BN to measure students' study progress and give the right assessment [4].

An ITS sometimes is an educational game, and it is provided by an intelligent agent that is done by Cristina Connati. Connati uses an intelligent agent for detecting the

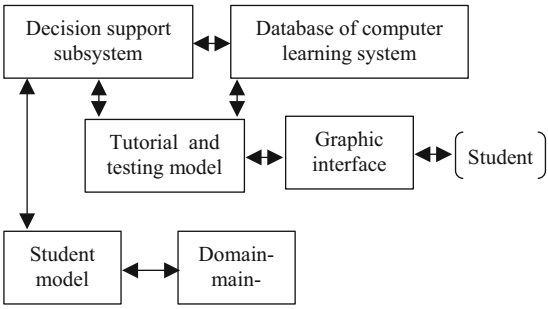

Fig. 1. The architecture of *ITS* [7]

interaction between a user and the educational game itself, and then it intervenes in the game to stimulate students in the learning process [5]. The [5] also applies socially an intelligent agent in a multiplayer game to monitor interaction among users. This monitoring process is useful for observing user emotions so the learning process becomes effective. In this case, Bayesian Network is used for modeling users or students. Bayesian Network is generally used for ITS because it can analyze the nondeterministic domains.

The students' or users' modeling is very important in ITS for determining the next lesson. Jim Reye used Dynamic Bayesian Network (DBN) for modeling students which are used as the backbone to link every BN [1].

Pedagogic modeling that was done by Rhania Hodod in her doctoral thesis [6] said that in Intelligent Tutoring System is close to motivation, mood, and cognitive process. The role of the pedagogic model is to adjust the instructions so the purpose of learning is achieved optimally that also used a Bayesian Network.

The general architecture of ITS is taken from Kaklauskas which is shown below.

Figure 1 shows the general architecture of ITS which has six modules. Tutor and testing model contributes to giving the tutorial and test to the users (students) through Graphical Interface. The tutorial or test that was given has been processed so that it is suitable to the user's requirement. This process is influenced by four other modules, there are; Decision Support Subsystem (DSS), Database of Computer Learning System, Student Model, and Domain Model. DSS is an interference machine thus ITS can give the precision decision that is reckoned by data from the database. The student model is very important because it is used for modeling a student, for example, achievement, learning style, moreover the other condition that might be detected as motivation, and others. The domain model also called Knowledge Domain contains the lesson information that will be taught.

Various ITS that have been developed and described above, Sunandan Chakraborty [8] said that those ITSs have a major drawback in the domain model that they are dependent on the knowledge domain itself. For example, ANDES that is developed by Conati is only compatible with newton's physic. If this system is applied to the other knowledge, it needs major changing. Furthermore, the student model and teaching model are dependent on the designer that must consult an education expert (expert system). For example, the different teachers will give a different method for a lesson. Therefore, it will produce a different ITS.

ITS that has been made by Chakraborty for resolving those drawbacks still has many shortages. First, the student model that has been built only covers two aspects, which are comprehension-ability and problem-solving skills. Secondly, this system had only been tested for one lesson that is C programming and it wishes that can be easily configured for the other complex lessons. On the other hand, Michael Heilman [9] said that ITS for language lessons is unique because it has a complex and wide knowledge domain.

This research will be developed an ITS that refers to the Chakraborty system by using the Bayesian Network as the main inference machine and it is applied to two related courses to compare those study paths.

2.2 Bayesian Network

This research is based on the Bayesian Network which is an extension of the Bayes theorem. Bayes theorem formulates a probability that is influenced by the other probability or called conditional probabilities.

The conditional probability of an event is the probability that is got from the information from the events that had happened before. Probability A because of B is symbolized by P(A|B) [10]. Determine the conditional probability is formulated by Thomas Bayes that is shown as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

For example, somebody that had come out from the train said that she had an interesting conversation with somebody, the probability of 'somebody' being a woman is 50% so it can be written by P(A) = 50%. From the next topic, it is got the information that 'somebody' had returned from the beauty salon. The fact that returning from the beauty salon strengthened that 'somebody' is a woman. If 'somebody' is symbolized by A and returning from the beauty salon is B so it can be written as P(A|B).

The bayesian network or Belief Network is a family of probabilistic graphical models [10]. This graphical structure is used to describe the knowledge of a nondeterministic domain. Bayesian Network is a triplet (N, E, P). N is a set of nodes each node is labeled by a random variable that associates with space. Each label is unique, so the definition of node and variable can be changed. E is a set of arcs, so D = (N, E) is Directed Acyclic Graph (DAG). This arc indicates the existence of direct causal influence among the variable. For each node Ai ∈ N, the strength of causal parent node πi is determined by Joint Probability Distribution (JPD) P(Ai|πi) from Ai that had been conditioned on the values of parent Ai. The base of dependency assumption that is attached to the Bayesian Network is an independent variable of its non-descendant that had been given by the parents. P is a Joint Probability Distribution. Bayesian Network with n node, P can be determined by this factorization.

$$P = p(A_1, \ldots, A_n) = \prod_{i=1}^{n} p(A_i|\pi_i) \tag{2}$$

For example, the JPD of the Bayesian Network is shown as follows: p (x1, x2, x3, x4, x5) = p(x1) · p(x2) · p(x3|x1) · p (x4|x1, x2) · p(x4|x5).

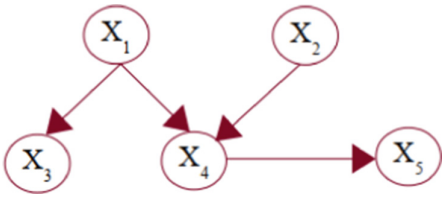

Fig. 2. Example of Bayesian Network and the *JPD* equations

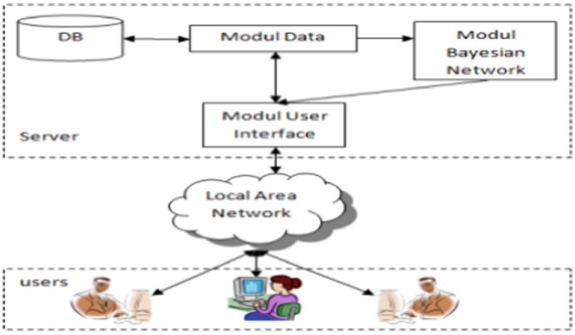

Fig. 3. Block Diagram of *ITS*

3 Research Method

3.1 Student Model Design

The data that has been used to construct a student model that describes student achievement as the result of the studying process is known by using the result of the chapter test. The relation between subject and questionnaire is structured by using Bayesian Network. Constructing BN as a student model on ITS refers to the information that is obtained from an expert (expert system) [11]. After BN for the student model has been constructed, the probability of each node is determined by using the data. Finally, this student model is ready to use for diagnosing a student of his understanding of a topic or knowledge component. In the student model, many factors such as student achievement, student weakness in a lesson, or others that are closed to the study process can be known by BN inference.

In the implementation step, the conceptual block diagram (see Fig. 2) is implemented really by using many modules, there is Database (DB), Data Module, Bayesian Network Module, and User Interface Module. Figure 3 shows ITS modules in this application.

3.2 Application Development

The application of ITS for Object Oriented Programming (OOP) that is developed here is based on Integrated Development Environment (IDE) Netbeans 7.3.1 as Java IDE programming language. Because this application is run online in Local Area Network

(LAN) so the Java Server Page (JSP) that is part of Java Enterprise Edition (JEE) technology is used. Bayesian Network inference machine that is built on server side uses the JavaBayes library [12].

3.3 Application Testing

The Application testing is done by trying it on a group of students and the system behavior will be observed as a response to every student's behavior. Therefore, it will show the difference in teaching methods among the students. The difference in studying methods is known by measuring the distance of every student's study tracks. The distance calculation is used Levenshtein Distance [13]. This testing is also done for comparing OOP lessons and Basic Java Programming. Besides that, The opinion of users about this application is determined by the Mean Opinion Score (MOS) that can be collected by a questionnaire that had been given to the users.

4 Result and Discussion

4.1 Implementation of ITS

This implementation is begun by designing every module that is needed. This planning phase is divided into two parts, there is inference engine design, and support modules design. The Planning of inference engine is purposed to produce Student Model and Knowledge Domain. Whereas support module planning is purposed to produce the design of user interface and database modules.

This design is started by collecting data for determining the Student Model and the Knowledge Domain. This collecting data phase needs data from the student study result of classical (classroom teaching) OOP training and Basic Java training. All of the students have not taken the OOP lesson and Basic Java programming lesson yet. After that, students are tested to determine their ability that is used for constructing a student model. Besides that, the student model and knowledge domain are also constructed by expert consideration that has experience in teaching those subjects [11]. This system is usually called by expert system. Therefore, researchers have observed many students that have taken this subject and lecturers that have experience in teaching this subject.

After data has been collected, the student model of Basic Java and OOP are constructed which are shown in Fig. 4. Before those student models are implemented on the system, those are simulated first in the Bayesian Network tool. In this case, BNJ is used that can be downloaded freely from bnj.sourceforge.net under GNU general public license (GPL) and it was developed by Dr. William H. Hsu et al. from Kansas State University. This simulation is done to ensure that those student models running well.

The student model for basic programming is the part of OOP tutorial because it is the prerequisite subject of OOP itself. For example, BN in Fig. 4 shows that the subject Array is influenced by Primitive and Object data types. It is because an Array is a data type that is a list of primitive or object data types. The probability of stressfulness of object data type is 68.7% and the primitive data type is 62.7%. Moreover, the *JPD* of the Array that is taken from the data is shown in Table 1 as follows.

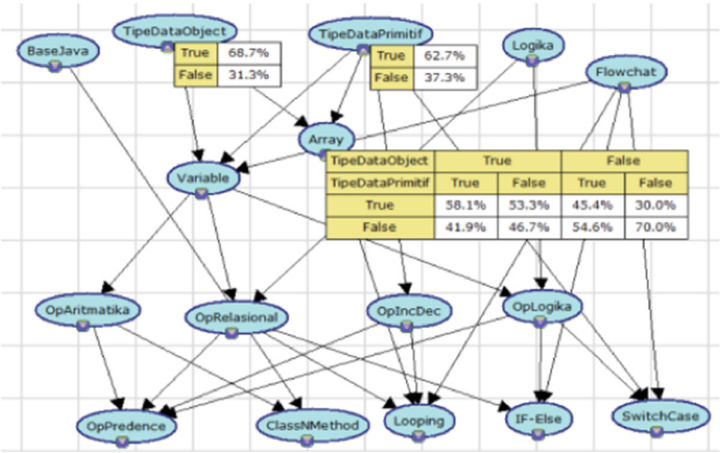

Fig. 4. Bayesian Network for Student Model of Basic Programming

Table 1. JPD of Array topic.

Object Data Type (O)	Primitive Data Type (P)	Array Data Type	P (A | O, P) %
T	T	T	58.1
T	T	F	41.9
T	F	T	53.3
T	F	F	46.7
F	T	T	45.4
F	T	F	54.6
F	F	T	30
F	F	F	70

$$\begin{aligned} p(AT) &= p\,(AT|PT,\ OT)\ \cdot\ p(PT)\ \cdot\ p(OT) \\ &+ p(AT|PF, OT)\ \cdot\ p(PF)\ \cdot\ p(OT) \\ &+ p(AT|PT, OF)\ \cdot\ p(PT)\ \cdot\ p(OF) \\ &+ p(AT|PF, OF)\ \cdot\ p(PF)\ \cdot\ p(OF)\ \cdot\ p(AT)\ =\ 51.1 \end{aligned}$$

Data from Table 1 that is applied in formulas (2) that are shown above will produce the probability for the array is 51.1%. It means that the number of students who understand is 51,1%. Analog to the example above, the student model of the OOP course is shown in Fig. 5 as follows.

The knowledge domain is constructed and stored in a database that contains many subjects, questions, and the relation among the subjects. It is used by the user interface to

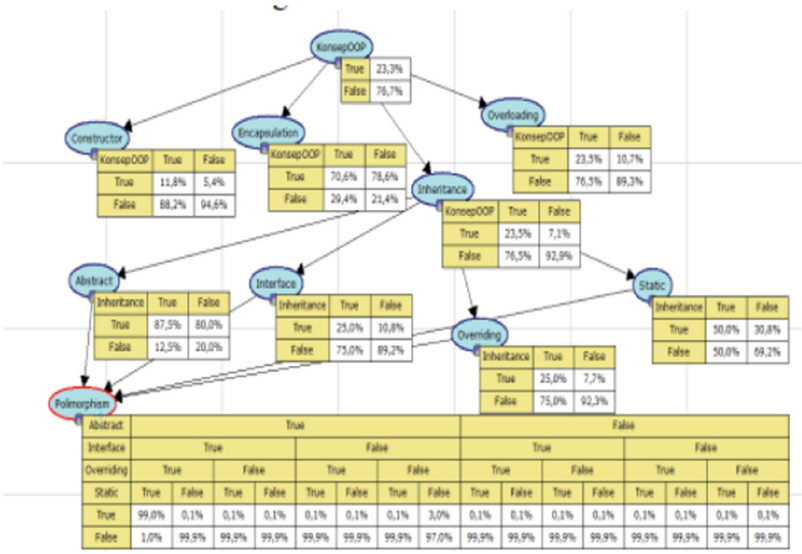

Fig. 5. Student Model and JPD of OOP course

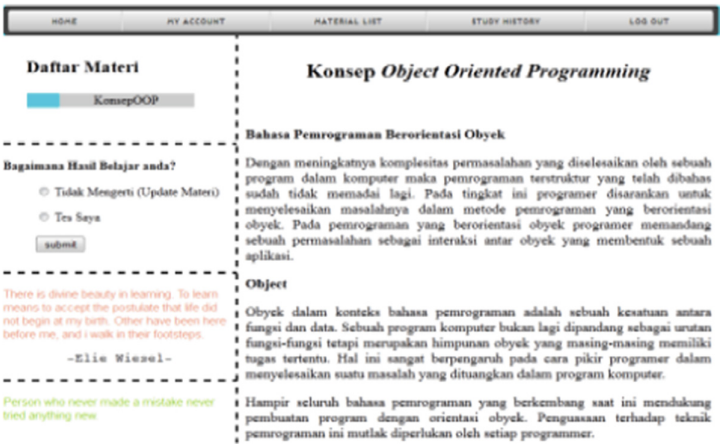

Fig. 6. Example page of Intelligent Tutoring System

conduct a tutoring process that is built as a web-based application by using *Java Server Page* (*JSP*).

After the user success to enter the system through a log-in page, there are available two choices that are intelligent tutoring or a conventional tutoring system that gives the materials in order. If the user chooses intelligent tutoring, then the user will find the page that is shown in Fig. 6 as follows.

Figure 6 shows a tutorial on the right part of that page, in the left part is shown two options there are ‘do not understand’ and ‘test me’. If the user chooses ‘do not

Fig. 7. Example of Evaluation page

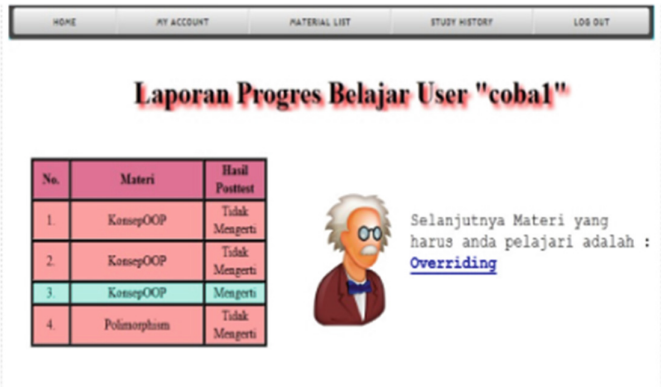

Fig. 8. The report page and the next lesson suggestion

understand, it means that this user is failed in this lesson. And then, by the student model, the system reckons the next material the student model. If the user chooses 'test me' the system will give the user the test page that is presented (Fig. 7).

The result of the reckoning process is a report of the studying process that is depicted in Fig. 8. It shows that a user failed in the lesson on the *OOP* concept, and then the second time he/she is given the same lesson. It is because the *OOP* concept is the root of the student model that has been shown in Fig. 5, so there are no other choices. However, on the third try, he passed this lesson, and then the system gave him the lesson on *Polymorphism*, and he failed again. After that system gave the lesson of overriding. It shows that this tutoring system is different from the conventional one because, in the conventional tutoring system, materials are presented in order. Therefore, if the user is failed in a lesson, then the system will repeat that lesson until the user passes it.

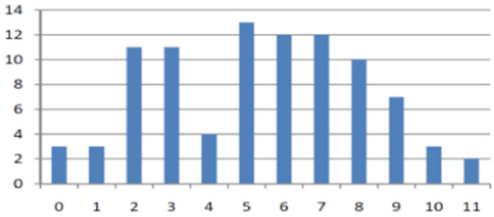

Fig. 9. Frequency distribution of the distance of the Basic Programming study path

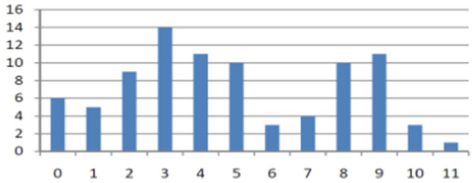

Fig. 10. Frequency distribution of the distance of the *OOP* study path

4.2 Differences in Learning Path

The difference in learning path among students is managed by the ITS that is following the achievements of students that is modeled by the Bayesian Network. Therefore, by measuring this difference is obtained the system performance that behaves as an human private teacher and gives appropriate materials to the student. This difference in learning style is measured with string distance Levenshtein. Of the 13 students that have tried basic programming applications, the shortest distance is 0 and the longest is 11. Figure 9 shows the frequency distribution of that data.

It is similar to the frequency distribution of Basic Programming, the frequency distribution of the distances of OOP is shown in Fig. 10.

From the graphs shown in Figs. 9 and 10, it can be seen that those distributions are not normal. However, in basic programming, its distribution is skewed to the right and OOP distribution is skewed to the left. This skewness can be calculated by Pearson's formula. This fact means that OOP distance is shorter than basic programming. In the other words, the basic programming tutoring system has a larger variation in learning path than OOP's learning path.

By the average of distances is 5.385, the median is 6, and the standard deviation is 2.72, the skewness of frequency distribution of basic programming is −0.68 or it has skewed right.

$$SK = \frac{3\left(x' - m_e\right)}{\sigma} = \frac{3(5.385 - 6)}{2.72} = -0.68$$

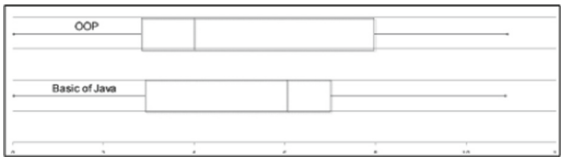

Fig. 11. Box plot of *Basic Programming* and *OOP* distances data

On the other hand, in OOP, the average is 4.8, the median is 4, and the standard deviation is 2.96, the skewness of OOP is 0.81 or skewed left.

$$SK = \frac{3\left(\overset{'}{x} - m_e\right)}{\sigma} = \frac{3(4.8 - 4)}{2.722.96} = 0.81$$

This data can be presented in a box plot that is shown in Fig. 11.

Figure 11 shows that Basic Programming data has a median value of 6 and largely data is concentrated in the third quartile. Besides that, largely data on OOP is concentrated in the second quartile. From both, the box plots can also be concluded that the distance of Basic Programming is further than OOP. This is normal because the basic course is usually easier than the advanced course.

4.3 Mean Opinion Score

To measure user opinion about this ITS, it is used Mean Opinion Score (MOS). The questionnaire used contains six questions, those are access speed, the response to the study report, user-friendly user interface, appearance, and the comparison of classical tutorial system and ITS. Each question has nine choices, there are (1) very disagree, (2) very disagree, (3) disagree, (4) less than enough, (5) enough, (6) more than enough, (7) agree, (8) very agree, and (9) very agree. And the result of this survey is shown that access speed, report responses, user friendly, and appearance have an 8 score, which means that the average opinion of students is very agreed. Only the stability aspect got 7 scores or agree. Besides that, in the comparison between ITS and classical tutoring, ITS has an 8 score and classical tutorial has a 7. Therefore, ITS is better than the classical tutoring system.

5 Conclusion and Suggestions

5.1 Conclusion

1. This ITS of OOP is run well that shown by the learning variation for each student that appropriates the Bayesian Network student model. By using the Levenshtein distance, it is calculated that the shortest distance of each variation is 0 and the longest distance is 11 with an average is 5.385 for Basic Programming and 4,8 for OOP.

2. OOP has positive skewness of distances frequency distribution that is 0,81. It means that the distance of learning paths in this tutoring system is generally shorter than in Basic Programming which has a skewness is −0,68 (negative) or skewed right. It is also shown in the box plot that the distances of OOP are concentrated in the second quartile with a median of 4, whereas in Basic Programming with a median of 6, the most data is gathered in the third quartile or it has further distance than OOP. This data shows that the variation of study paths in basic programming is larger than OOP.
3. The result of MOS shows that most students are more interested in using intelligent tutoring rather than classical tutoring the average point of ITS is 8 (very agree) and classical tutoring is 7 (agree). Whereas access speed, stability, user-friendly, report responses, and appearance generally get 8 points or very agree, and only one aspect that is stability has 7 average points or agree.

5.2 Suggestions

In future work, it is needed to test the accuracy of the provision of material that depends on the structure of the Bayesian Network in the Student Model. Besides that, It needs testing in a large number of users and also in the more varied subject. Moreover, the comparison of the performance of the Bayesian network as an inference engine with other inference engines is also needed.

References

1. Reye, J.: Student modelling based on belief networks. In: School of Information System, Queensland University of Technology, Brisbane, Australia, j.reye@qut.edu.au, Int. J. Artif. Intell. Educ. **14** (2004)
2. Ioannis, H., Jim, P.: Using hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities. In: Expert System with Application. Elsevier (2003)
3. García, P., Amandi, A., Schiaffino, S., Campo, M.: Using Bayesian networks to detect students' learning styles in a web-based education system. In: 7th Simposio Argentino de Inteligencia Artificial—ASAI2005, Rosario (2005)
4. West, P., et al.: A Bayesian network approach to modeling learning progressions and task performance. In: National Center for Research on Evaluation, Standards, and Student Testing Graduate School of Education & Information Studies UCLA | University of California, Los Angeles (2010)
5. Conati, C., Klawe, M.: Socially intelligent agents in educational games. University of British Columbia. http://www.cs.ubc.ca/spider/conati/
6. Hodhod, R.: Interactive narrative for adaptive educational games: architecture and an application to character education. University of York, Artificial Intelligence Group Department of Computer Science United Kingdom (2010)
7. Kaklauskas, A., et al.: Intelligent Tutoring system for real estate management. Int. J. Strateg. Prop. Manag. (2006)
8. Chakraborty, S., Roy, D., Basu, A.: Development of knowledge-based intelligent tutoring system. In: Sajja, Akerkar (eds) TMRF e-Book Advanced Knowledge-Based Systems Chapter 5: model. Applications & Research, vol. 1, pp. 74–100 (2010)

9. Heilman, M., Eskenazi, M.: Language learning: challenges for intelligent tutoring systems. Language Technologies Institute, Carnegie Mellon University, mheilman@cs.cmu.edu (2006)
10. Ruggeri, F., Faltin, F., Kenett, R.: Bayesian networks. In: Encyclopedia of Statistics in Quality & Reliability. Wiley (2007)
11. Duijnhoven, J.: Knowledge assessment using Bayesian Networks A case study in the domain of algebraic expectation. Utrecht University (2003)
12. Cosman, F.G.: Bayesian Networks in Java. University of São Paulo (2001). http://www.cs.cmu.edu/~javabayes/Home/index.html
13. Hyyrö, H.: A bit-vector algorithm for computing Levenshtein and Damerau edit distances. Department of Computer and Information Sciences 33014 University of Tampere, Finland