Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 35 (2008) 677–685
Expert Systems with Applications
www.elsevier.com/locate/eswa

Verification and validation of an intelligent tutorial system

R.M. Aguilar [a,*], V. Muñoz [a], M. Noda [b], A. Bruno [b], L. Moreno [a]

[a] *Dpto. Ingeniería de Sistemas y Automática y Arquitectura y Tecnología de Computadores, Universidad de la Laguna, Spain*
[b] *Dpto. Análisis Matemático, Universidad de la Laguna, Spain*

Abstract

This paper presents the results of a verification and validation process for an intelligent system. The system being studied is an Intelligent Tutorial that employs fuzzy logic and multiagent systems. Software engineering techniques were used in the verification process, while the validation exploited both qualitative and quantitative techniques.

Keywords: Verification and validation; Fuzzy system; Multiagent system

1. Introduction

One of the most important and difficult tasks facing a knowledge engineer is the verification and validation of an intelligent system (Mosquera & Moret, 2001). The experts who use the system outputs to aid in making recommendations and managers who make decisions based on these recommendations justifiably look upon an intelligent system with a certain degree of scepticism about its validity. It is the job of the knowledge engineer to work closely with the end users throughout the development and validation period to reduce this scepticism and to increase the system's credibility (Domingo, Martín-Baranera, Sanz, Sierra, & Uriz, 1999). The goal of the validation process is twofold:

(a) to produce an intelligent system that represents true system behaviour closely enough so that the system can be used as a substitute for the actual expert for the purpose of aiding in decision making,
(b) to increase the credibility of the intelligent system to an acceptable level so that users can rely on the system in their decision making.

Validation should not be seen as an isolated set of procedures that follows intelligent system development, but rather as an integral part of the system development cycle. Conceptually, the verification and validation process consists of the following components (Sargent, 1998):

- Verification is concerned with building the intelligent system correctly. It is utilized to compare the expert decision making to that of the computer representation that implements that decision. It asks the questions: Is the intelligent system implemented correctly in the computer? Are the input parameters and the logical structure of the intelligent system correctly represented?
- Validation is concerned with building the correct intelligent system. It is utilized to determine whether or not a system is an accurate representation of the knowledge of the experts. Validation is usually achieved through the calibration of the intelligent system, an iterative process of comparing the intelligent system to expert behaviour and using the discrepancies between the two, and the insights gained, to improve the intelligent system. This process is repeated until system accuracy is judged to be acceptable.

This paper describes the method that has been used in the verification and validation of an intelligent system that models a teacher in the teaching-learning process. The

* Corresponding author.
E-mail address: raguilar@ull.es (R.M. Aguilar).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2007.07.024

behaviour of this system is defined by strategies which adapt the learning process for individual students by applying appropriate pedagogical methodologies. For this reason, we propose the combined use of a fuzzy system and a multiagent system. The first models the uncertain knowledge of the expert teacher, while the latter implements the way to assign the main objectives of a learning system, in order to efficiently teach the student.

This paper is organized as follows. In Section 2, the intelligent system is described. Next, we present the verification process employed. Section 4 shows the system's qualitative and quantitative validation. The paper ends with a summary of the results.

2. Its for reinforcement addition

The aim of this project was to develop an intelligent system for teaching and reinforcing the basic concepts of a number, addition and subtraction for kindergarten and primary-school students. The individualized teaching process involved determining the learning goals based on each student's characteristics. Then a series of activities to be performed by the student was devised which allowed him to become acquainted with the concepts or acquire objective-based knowledge. There was not one set of standard activities for every student to reach the same objective; rather, these depended on each student's characteristics. That is, a set of learning objectives was developed for each student and a series of actions devised to reach those objectives.

The intelligent system, therefore, must be a dynamic module capable of generating a plan of activity, of monitoring its execution and of re-planning when necessary. Due to the characteristics of the educational knowledge to be modelled, this tutorial was designed using fuzzy logic and multiagent systems. In the proposed design, then, once the student's learning objective and the results of current (student state) and past (student history) actions are known, the intelligent system decides on the set of activities to be assigned to the student to reach the goal, as well as their level of difficulty.

The proposed system is made up of four basic components (Fig. 1):

- *Domain module:* stores the concepts to be taught and the possible activities to assign. It is implemented via a database.
- *Student module:* stores the student-specific information: physical age, cognitive age, motor functions, etc. It also uses a database in its implementation.
- *Tutor module:* charged with planning the tasks assigned to a specific student based on that student's data. It is implemented by the combined use of fuzzy logic and a multiagent system.
- *Interface module:* its purpose is to present the various activities to the student for him to solve. It is implemented by using XML templates which are displayed like web pages.

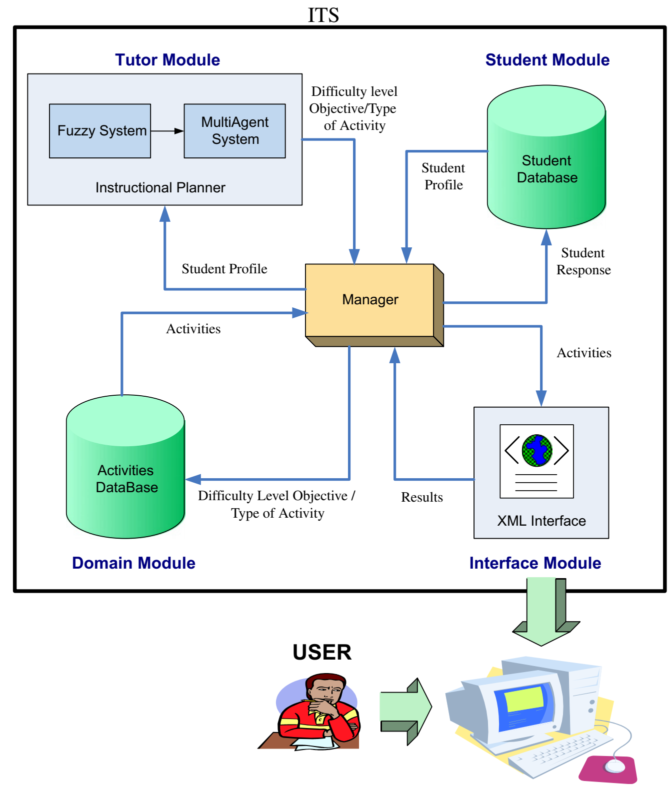

Fig. 1. Intelligent tutorial structure.

The ITS follows this sequence:

1. The manager requests information from the student model about the personality as well as information about his/her learning progress.
2. The manager sends these parameters to the Instructional Planner (fuzzy and multiagent system) and requests the difficulty level, the next objective and the kind of activity to be performed by the student.
3. The manager searches for activities in the database that match this difficulty level.
4. The chosen activities are sent to the multimedia interface which generates a Web page. In this Web page, a pedagogical agent presents the activity to be carried out by the student. When the task is finished, the multimedia interface sends the results to the manager, and are then stored in the database.

3. Verification

Testing is an activity to verify that the correct system is being built. Testing is traditionally an expensive activity since many faults are not detected until late in the development process. A qualitative and well-organized approach to system development is necessary in order to increase the quality of the system and to lower testing costs. To do effective testing, every test should aim to detect a fault. There are several different types of tests and testing techniques, some of which have been used in the intelligent tutorial. In order to verify this intelligent system we have used testing levels: unit testing, integration testing and system testing (Jacobson, 1997).

3.1. Unit testing

Unit testing means that one, and only one, unit is tested as such. To this end, the intelligent system was divided into three main parts: graphic interface, fuzzy system and multiagent system. This verification was done by knowledge engineers who used specification, or black-box, testing, and structural, or white-box, testing. The former is used to verify the input/output relationship of each unit. The goal is to verify the specified behaviour of the graphic interface, fuzzy system and multiagent system; that is, to verify what each unit does, but not how it does it.

The purpose of structural testing is to ensure the internal structure is correct. This test is also called program-based testing. Knowledge engineers use their knowledge of how the unit was implemented when they test the graphic interface, fuzzy planner and multiagent system. The procedure is based on covering all possible combinations of parameters, variable values and paths in the code during the testing. Since the structural test cases are dependent on the code structure, and the specification testing may modify this code structure, the structural testing was done last.

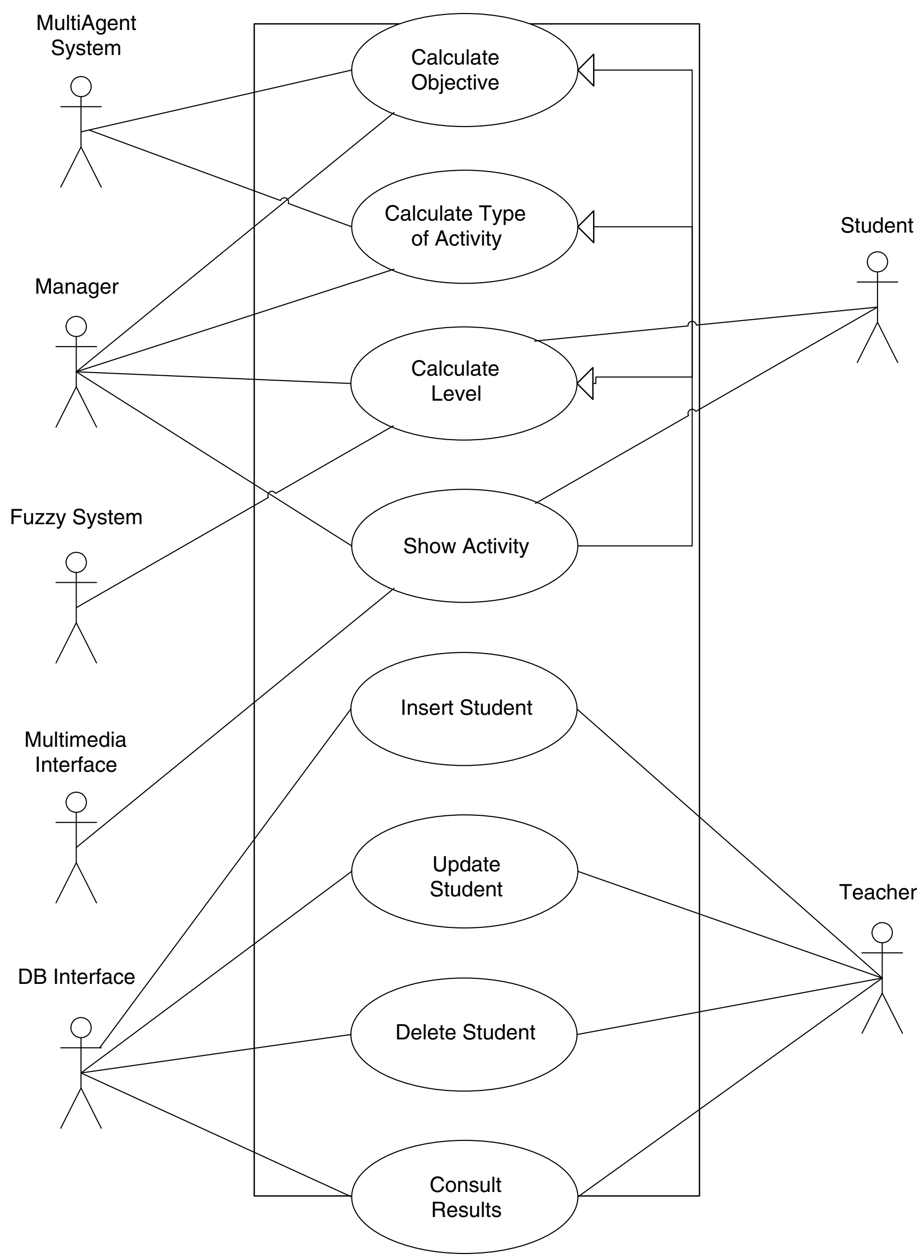

Fig. 2. Intelligent system use-case diagram.

3.2. Integration testing

Integration testing involves verifying that the units are working together correctly. For this we employ use-case diagrams, Fig. 2. This diagram graphically shows the interactions between the participants (teachers, student, and intelligent tutorial), describing all the actions performed by each (Table 1).

3.3. System testing

System testing involves verifying the application as a whole. This takes an end-user view of the system and the cases perform typical end-user actions. This testing was done by the knowledge engineers and expert teachers. Here, the system is tested in normal operation for a longer time. The Intelligent Tutorial is used in the intended manner. Only normal mistakes are made, that is, mistakes that the normal user may be expected to make.

Since the intelligent system considers three different users (motivated, afraid of failure and hyperactive), each type was simulated. This was done because the sequence to be followed by the intelligent system depends to a large extent on this characteristic. The information flow was as indicated by the expert teachers. The simulation of a student afraid of failure performed many more activities before going on to the next phase, while hyperactive students went much faster, although they readily went back to previously studied phases. The case of a motivated student was somewhere in between.

In addition, the experts devised ergonomic tests which showed how ITS come to be used by non-computer professionals, that is, by the students. The man–machine interface was verified by means of the following questions:

(a) Are the menus logical and readable? It was confirmed that a web layout facilitated the use of the tutorial since it was familiar to the teachers.
(b) Are system messages visible? In this regard the audio mechanism had to be changed. Initially, a text-to-speech system was used but the students did not understand the text. It was then decided to have the teachers record the different sections.
(c) Are the failure messages understandable? Once the audio was changed as specified above, the messages did not cause any problems.
(d) Does the system provide the same conceptual picture that the end user has? The students did not understand certain images, so the tasks were contextualized and complicated graphics were deleted.

4. Validation

An intelligent system may be logically correct without being valid (Mosquera & Moret, 2002a; Zlatareva, 1998). Validation has to do with how well a model conforms to what has been modelled. Thus, validation has to examine the results of the intelligent system, and check if the Intelligent Tutorial meets the user's requirements. In order to validate, the performance of the Intelligent System has to be compared with the desired performance as provided by the experts, both teachers involved in the project and independent teachers. To do this we used both qualitative and quantitative validation methods. The qualitative methods used subjective techniques to determine the system's performance and usability, while the quantitative used statistical measures. The combination of both yielded the best results.

Table 1
Functionality for each use-case

Use-case	Participants	Purpose	Overview
Calculate level	Student, manager, Fuzzy system	Calculate an objective's difficulty level	Once the student has completed an objective's activities, the manager activates the fuzzy system and sends it the data so it can calculate the next level to assign to the student in that objective
Calculate objective	Manager, multiagent system	Calculate the next objective to be assigned to the student	To determine the next objective to be assigned to the student, the manager activates the multiagent system and sends it the student's information. The system then determines the next objective based on these data
Calculate type of activity	Manager, multiagent system	Calculate the type of activity to be shown to the student	Once the next objective and the level to be assigned to the student have been determined, the manager activates the multiagent system and sends it the student's information. The multiagent system determines the type of activity to be assigned to the student for the given objective
Show activity	Student, manager, multimedia interface	Show the student the activity (or activities) to be solved	Once the manager knows the objective, the level for the objective and the type of activity within the level corresponding to the student, he selects one or several activities to be shown to the student via the multimedia interface
Add student	Teacher, DB assistant	Add a new student to the tutorial	The teacher uses the Data Base (DB) assistant to input the new student's personal data into the application
Modify student	Teacher, DB assistant	Modify student information	The teacher uses the DB assistant to modify the data of a student already in the application
Delete student	Teacher, DB assistant	Delete a student from the Tutorial	The teacher uses the DB assistant to delete a student from the application
Check results	Teacher, DB assistant	Check the results obtained by a Student	The teacher uses the DB assistant to check the results obtained by a student for the objectives in each phase of the tutorial

The qualitative validation was performed on the entire system, while the quantitative validation was performed on the two key parts of the intelligent system (the fuzzy and multiagent systems). In order for the final results to be correct, the intermediate results have to be evaluated, since the former depend on the latter. In this way the system's method for making inferences is validated, and thus if the correct answers are output, we know it was not as a result of chance or luck.

4.1. Qualitative validation

The qualitative validation used subjective techniques to determine the system's performance and usability. In our case it was carried out by the two main types of users:

- the teachers at the centres that are going to use the intelligent tutorial, and who are experts in the subject matter and therefore valid users when testing the application (field test), and
- kindergarten and primary school students for whom the intelligent system is intended, who obviously are not experts in the subject matter and therefore take part in validating the part corresponding to the user.

The results were videotaped over several sessions. The teacher was seated next to the child during the sessions in case the child needed help operating the computer.

Regarding the student's ease of use with the intelligent tutorial, two key aspects were noted in the students' approach to the computer: the use of the mouse and his self-reliance when performing the activity.

4.1.1. Use of the mouse

The tutorial makes use of two actions with the mouse: clicking and moving. To facilitate the use of the mouse, the dragging function was modified so that students only have to click on an image, which then becomes the mouse pointer, move it and click again to release it in the desired place. The results seen in the recordings revealed that some students had problems using the mouse, (Fig. 3). This was due especially to the low exposure these students had to computers at home and school (in fact, for some students it was their first time using a computer). Sometimes the students verbally solved the exercise but failed to solve the activity when using the computer; in other cases, they pointed to the result on the screen with their fingers. As a result of this, we deemed it necessary to incorporate the use of touch screens to facilitate the use of the tutorial for those students who so require it.

4.1.2. Self-reliance

As for the students′ self-reliance with the computer, a key aspect to consider was whether the students were able to handle the activity by themselves or required the aid of the teacher. While the results vary, it was noted that no student was able to operate the computer without help, Fig. 3. It should be noted that these students were used to working in a very interactive environment and that in some cases they had difficulty understanding the activity statements and the required actions. Text-to-speech technology was used at first, that is, using a synthesized voice to read the text. This allowed the intelligent system to reproduce text input into the computer. However, due to the low degree of self-reliance exhibited by the students, it was decided to record the voices corresponding to the activities' explanations, feedback, etc., and play them during the tutorial.

4.2. Quantitative validation

The quantitative validation employed a series of cases (observations) for both the fuzzy and multiagent systems,

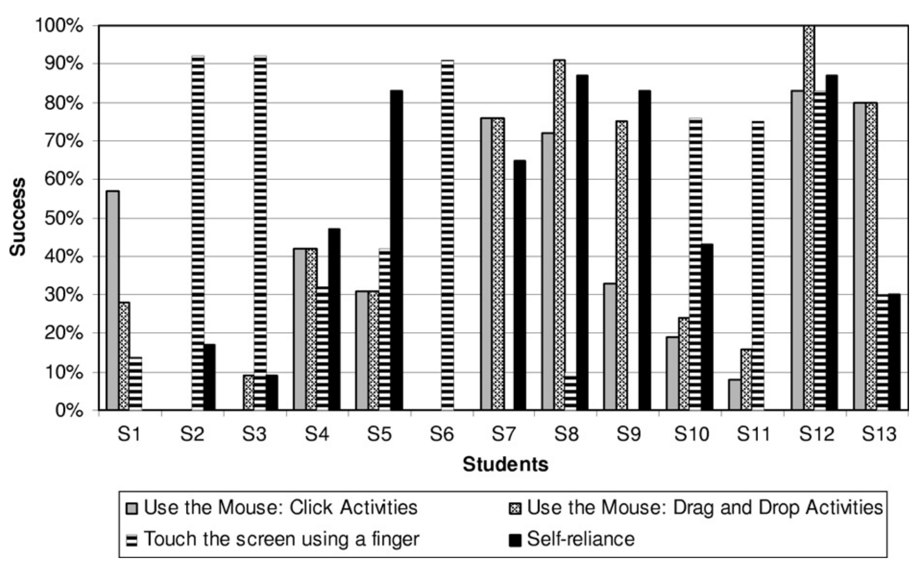

Fig. 3. Results of the student interactions with the tutorial.

such that the results obtained during the tutorial simulation could be compared with the results proposed by the experts for said cases. This series of cases must meet two essential criteria for the sample to be significant: quantity (the number of cases must be significant) and representability (the cases must cover the range of the application). These cases were validated by a group of experts. Such a technique has the added advantage of offering opinions from several experts (instead of only one or a consensus opinion), which leads to fewer errors in the results and also allows the degree of agreement or similarity between the experts' answers to be compared.

One example of a case for the fuzzy system validation would be: Given a hyperactive student doing medium-level activities, and considering his low performance with such activities in the past, if when performing the new activities he gets 50% of them right, what should the next difficulty level assigned to the student be? Should he be kept at that level, be dropped to the next lower level, or be introduced to the next higher one?

The statistical measures we used for the quantitative validation were pairs' methods, in which the results of the tutorial are compared with each expert's interpretation for the cases being considered. These pairs' methods can be grouped into three types: measures of agreement, of association and of predictive association (Mosquera & Moret, 2002b).

The measures of agreement provide a measure of the coincidence that exists between the interpretation of two experts, or of an expert and the tutorial system (Table 2).

The measures of association provide us with the degree of linear association that exists between the tutorial system and the expert. One of the characteristics of these measures is that they can only be applied in the case where the categories of interpretation for the results are ordinal, in other words, when a range can be given for ordering the possible results of the cases. In the intelligent tutorial system, they can only be applied for the case of the fuzzy system (Table 3).

Lastly, the predictive association measures let us determine the extent to which our system's interpretation can be used to predict the expert's interpretation (Table 4).

Validation is not an easy task. For this reason, we used a divide-and-conquer approach, dividing the instructional planner into two independent subsystems: a fuzzy system and a multiagent system. The multiagent system is tasked with determining the student's next objective (and the activity within that objective). The fuzzy system is used to determine the assigned objective's level of difficulty for that student.

Table 2
Measures of agreement used in validating the intelligent system

Index of agreement	Ratio of the number of observations where there is complete agreement between the ITS and the experts to the total number of observations. Its value is between [0, 1]. The closer it is to 1, the better the agreement
Kappa index	Measure of agreement which corrects for agreements due to chance. Obtained by deleting a percentage of the agreements expected by chance, this being the sum of the products of the marginal proportions corresponding to the main diagonal (the marginal proportions are the sum of all the values in a row or column). A value equal to 1 indicates perfect agreement. A value of 0 means the agreement is no better than what would be expected by chance
Weighted kappa	Measure of agreement that corrects for agreements due to coincidence and applies a different weight to the disagreements. It uses a weighted matrix where the existing disagreement is quantified such that the main diagonal is assigned a value of 0, indicating perfect agreement. This value increases with distance from the diagonal. (For example, it is more of a mistake for the system to output Very Low when the correct result is Very High than it would be if the correct result were Low.)

Table 3
Measures of association used in validating the intelligent system

Kendall Tau-b	Non-parametric measure of the correlation for ordinal variables or ranges which considers ties. The degree of separation between categories is not taken into account. The sign of the coefficient indicates the direction of the correlation, and the absolute value its magnitude, such that a large absolute value indicates a stronger correlation. The possible values go from −1 to 1
Spearman Rho	Correlation coefficient based on ranges. It is suitable for ordinal data that are invariable to transformations that maintain the order. It considers the order of separation between categories. A result between 0 and 1 is the same as Tau. The closer the value is to 1, the greater the correlation
Gamma	Measure of the symmetric association between two ordinal variables and whose value is always between −1 and 1. Absolute values near 1 indicate a strong correlation between the two variables. Values close to zero indicate little or no correlation between the two variables

Table 4
Measures of predictive association used in validating the intelligent system

Lambda	Measure of association that reflects the proportional decrease in the error when values of the independent variable are used to predict the values of the dependent variable. A value of 1 means the independent variable perfectly predicts the dependent variable. A value of 0 means the independent variable is of no use in predicting the dependent variable
Goodman–Kruskal Tau	Obtained using the principle of proportional error reduction. It is a measure of directional association. A value of 0 indicates no association, and 1 a perfect or complete association

4.2.1. Fuzzy system validation

The fuzzy system outputs a categorical ordinal variable that has three categories (regress, remain and progress). Within a specific objective, these represent lowering the activity difficulty for the student (regress), staying at the same level (remain) and increasing the difficulty level (progress).

The measures of agreement used in the validation techniques were: the index of agreement and the weighted kappa measure.

As for the measures of association, we will use the measures derived from Kendall's tau (the tau b and gamma) and Spearman Rho.

Fig. 4 shows a summary of the results of the validation for the fuzzy system. This system is used to determine the level of the next activity to be assigned to the student. It shows the results of the validation done with experts, both for those involved in developing the project (expert1–expert3) and those external to the project (expert4–expert15). It is interesting to note that all the values obtained were, on average, above 0.80. The Gamma statistic value close to 1 in every case shows a strong correlation between the experts' values and those obtained with the fuzzy planner.

4.2.2. MultiAgent validation

The MultiAgent System (MAS) can be divided into two subsystems:

1. MultiAgent System for Objectives (MASObj), which determines the next objective to be assigned to the student.
2. MultiAgent System for Activities (MASAct), which indicates the next activity type to be shown to the student.

The output variable for the MASObj is a categorical nominal variable that has as many categories as there are objectives within a phase. The MASAct also outputs a categorical nominal variable that has as many categories as there are types of activities within an objective.

For the validation we will use the indexes of agreement and the kappa index as the measures of agreement. The use of the weighted kappa index cannot be justified since no discrepancies are more serious than others among the categories. We will also study the predictive association variables to determine the extent to which the Multiagent system's interpretation can be used to predict that of the expert. Specifically, we will use Guttman's and Goodman–Krustal's lambda (λ) and the Goodman–Kruskal tau (τ).

4.2.2.1. MASObj system.

Fig. 5 shows the results of the system validation as compared with the experts. The results output by the MultiAgent System when choosing the next objective to assign to the student were compared against those decided on by the experts for a series of cases. Note that values above 0.80 were obtained for the indices of agreement, the value being somewhat lower for the predictive measures of association, although on average the results were above 70%.

4.2.2.2. MASAct system.

Fig. 6 shows the results for the validation of the Multiagent System when choosing the next activity to be presented to the student (the objective having been chosen previously). Note that, on average, all the results obtained were above 75%.

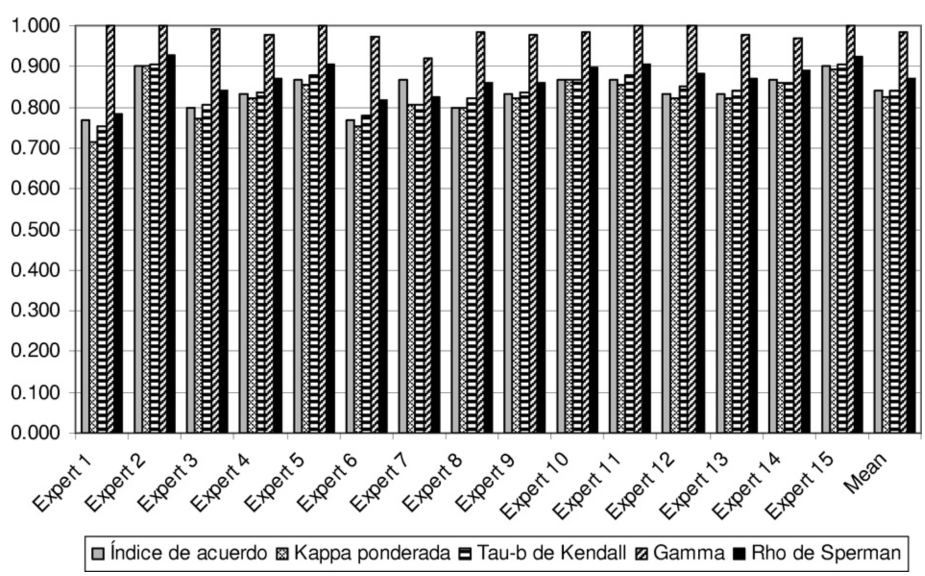

Fig. 4. Results of the fuzzy system validation.

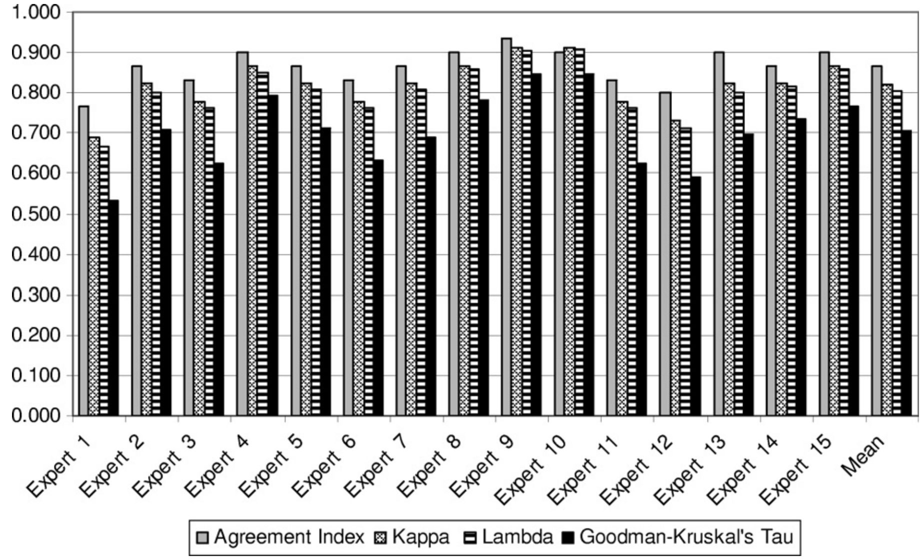

Fig. 5. Results of the validation for the objective multiagent system.

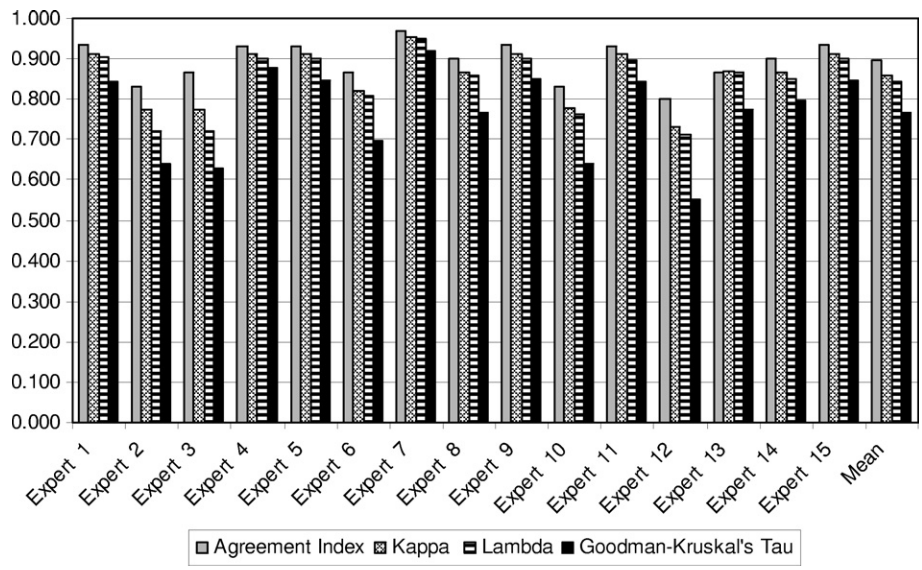

Fig. 6. Results of the validation for the Activity Multiagent System.

5. Conclusions

Verifying and validating intelligent systems is a very complex task. It is an essential one, however, if the Intelligent System is to be put into practice. In this paper, we have shown the verification and validation for an intelligent tutorial system intended to reinforce the logical concepts behind numbers, addition and subtraction. This task was performed by breaking down the tutorial into its component parts: the fuzzy system, the multiagent system and the multimedia interface. The verification process employed the techniques commonly used in software engineering. For the validation, both qualitative and statistical techniques were used. The former indicated the changes needed (not using text-to-speech, not dragging with the mouse, etc.) to improve the intelligent system's usability. The statistical techniques were used to validate the intelligent system's inference process by providing a measure of how close the answer given by the intelligent system was to that given by an expert teacher. It is worth noting that the verification and validation process performed on the intelligent tutorial has served to increase its credibility among teachers, who will now use it as one more tool in the classroom.

Acknowledgements

This work is being supported by an FPU Grant (Ref. AP2002-3850) from the Ministerio de Educación y Ciencia.

We are grateful to Asociación de Trisómicos 21 de Tenerife (ATT21).

References

Domingo, M., Martín-Baranera, M., Sanz, F., Sierra, C., & Uriz, M. J. (1999). Validation SPONGIA, an expert system for sponge identification. *Expert Systems with Applications, 16*.

Jacobson, I. (1997). *Object-oriented software engineering: A use case driven approach*. Addison-Wesley.

Mosquera, E., & Moret, V. (2001). Validation of intelligent systems: A critical study and a tool. *Expert Systems with Applications, 18*.

Mosquera, E., & Moret, V. (2002a). Intelligent interpretation of validation data. *Expert Systems with Applications, 23*.

Mosquera, E., & Moret, V. (2002b). Validación de sistemas inteligentes. *Tórculo Edicións*.

Sargent, R. G. (1998). Verification and validation of simulation models. In *Proceedings of the 1998 Winter Simulation Conference*.

Zlatareva, N. P. (1998). A refinement framework to support validation and maintenance of knowledge-based systems. *Expert Systems with Applications, 15*.