Validation and Usability Analysis of Intelligent Systems: An Integrated Approach

Mosqueira-Rey, Eduardo; and Moret-Bonillo, Vicente

Abstract— V*alidation of intelligent systems is a complex matter due to the lack of standard references for complex domains. Moreover, the validation phase should be followed by a usability analysis for studying the quality of man-machine interaction. The VISNU (Validation of Intelligent Systems aNd Usability) tool has been designed to assist developers in the validation and usability analysis phases in intelligent system design. The validation module includes quantitative measures (such as pair tests, group tests and agreement ratios) and facilities for planning the entire process and for interpreting the final results. The usability module includes different types of usability analyses, namely, heuristic (based on the collaboration of experts), subjective (based on the collaboration of users) and empirical (based on objective data). One of the main goals of the system developers has been to integrate different evaluation methods to obtain information which could not otherwise be obtained.*

Index Terms—*Key words or phrases in the alphabetical order, separated by commas*

1. INTRODUCTION

Like all computer systems, intelligent systems require, as part of its development methodology, the inclusion of a process for analysing the functioning of the system [1]. This process is usually divided into different phases that analyse particular aspects of the system. Although these phases are designated by a range of well-known terms (verification, validation [2], usability analyses, etc), definitions tend to vary widely among authors, and despite efforts to standardise the terminology [3], the reality is that each author tends to use his/her own definitions. Below is a list of the most commonly used definitions [4][5]:

- *Verification*: the process that ensures that the system is structurally correct.
- *Validation*: the process that ensures that the system results are correct.
- *Usability analysis*: the process that tests aspects related to the human-computer interaction (HCI).
- *Utility analysis*: the process that tests the benefits of the system in the domain within which it will be used.
- *Evaluation*: the process of performing an overall analysis that includes the above-mentioned phases.

These phases generally follow a logical developmental sequence. However, contemporary iterative and spiral development methodologies execute these phases in several cycles, with gradual increases in range.

1.1 Verification and Validation

Verification is a 'white box' analysis of the system; in other words, the internal structure of the system is analysed in order to uncover possible errors or anomalies. Boehm [6] defined verification as the process of checking whether we are "building the product right".

An important verification limitation is the fact that this phase involves an internal analysis of the system, with the implication that systems with different structures need to be verified using different strategies. Thus the verification of intelligent systems is not quite the same as the verification of conventional systems; moreover, it is not quite the same to verify an intelligent system based on production rules and one based on Bayesian networks, to just cite one example. What's more, in many cases verification depends on the specific tool used for coding the system.

Validation, meanwhile, consists of a 'black box' analysis of the system; in other words, it is not the internal functioning of the system which is being tested, but rather the responses of the system to a specific set of inputs. Boehm [6] defined validation as the process of checking whether we are "building the right product" according with the previously stated definition of validation.

There is an important implication in the fact that the validation phase considers the system as a black box: the models, methods and tools designed to support this phase can be applied to any intelligent system, since there is no need to take into account the internal structure of the

Manuscript received March 24, 2005. This work was supported in part by by CICYT-ERDF (Project TIC2001-0569). The authors are with the Computer Science Department, University of A Coruña Spain (e-mails: {eduardo, civmoret}@udc.es). The contact person is V. Moret-Bonillo).,

system or the tool used for its design. One of the first works containing an analysis of the validation phase was an article by O'Keefe et al. [7]. In this article, important questions in relation to validation were posed, for example: What do we evaluate? What data do we use for the validation? How do we evaluate? Who should be involved in the process? Further articles on the subject were subsequently published by O'Keefe and O'Leary [8] and by Gupta [9]. A description of the main verification and validation tools between the years 1985-1995 can be found in [10].

Of all the issues raised by validation, probably the most important question is in relation to the validation criterion. In other words, if validation treats the system as a black box and only analyses its outputs, then we require a criterion that will indicate whether these outputs are correct or otherwise. This poses no particular validation problems with conventional systems, given that it is a relatively simple matter to check whether an algorithm produces the expected results. However, for intelligent systems that model human knowledge, the identification of a validation criterion is more problematic. In practice, one of two approaches to validation are typically taken, defined according to the kind of criterion used:

- *Validation against the problem*: A standard reference exists that is used to test that the system results are correct.
- *Validation against the expert*: No standard reference exists and so system interpretations must be compared with those of human experts from the domain.

Validation against the problem is the more ideal approach; it generally relies on measures such as true/false positive/negative ratios, which are combined in graphs such as ROC curves (Receiver Operating Characteristic) [11].

Should this approach not be possible, then evaluation against the expert is the next best alternative. To avoid subjectivity in the validation process, it is recommended that several experts (not involved in the design of the system) be used. The ideal approach consists of a standard reference obtained by means of the experts reaching a consensus in their interpretations using a technique such as Delphi [12]. Nonetheless, the process of developing a consensus is both slow and costly, and so the normal procedure is for the experts to work individually. This has the advantage, however, of permitting an analysis of any inconsistencies that may arise in individual interpretations.

Another problem of validating an intelligent system against a group of experts is that the volume of information obtained is high, and this requires the use of statistical and multivariate analysis techniques to facilitate the interpretation.

1.2 Usability and Utility

Verification and validation have been performed in conjunction on so many occasions that they have jointly become known as 'V&V'. Nonetheless, more recently, usability and utility analyses have been attracting the interest of system developers, above all as a consequence of applications becoming accessible to individuals without computer knowledge, through networks such as the Internet and applications such as the World Wide Web [13][14].

Whereas verification and validation are concerned with system functioning, usability analyses endeavour to evaluate aspects that go beyond correctness of results and that involve the quality of the man-machine interaction. Utility analysis, on the other hand, rather than evaluating whether the system is usable, evaluates whether it can bring additional benefits to users. Although usability and utility are two distinct evaluation phases, in practice both are analysed jointly. Adelman and Riedel [4], for example, provide a questionnaire for a joint analysis of both phases.

There are many usability analysis techniques available, which various authors have classified in different ways. Preece [15], for example, classified usability analyses as analytic, expert, observational, survey and experimental. Another interesting work by Ivory and Hearst [16] includes a classification of usability analysis techniques and tools in terms of the four dimensions of method class, method type, automation type and effort level. The method class category (equivalent to Preece's classification) includes testing, inspection, inquiry, analytical modelling and simulation. In our research we have preferred the Adelman and Riedel classification, which identifies three types of techniques for analysing usability:

- *Heuristic*: These techniques are based on the opinions of usability experts, who analyse the system and determine strengths and weaknesses from an end-user perspective.
- *Subjetive*: These techniques are based on the opinions of the system users, who analyse operational prototypes and give their opinions on the usability of these prototypes.
- *Empirical*: These techniques, which are based on the actions of the system users, function on the basis of obtaining objective data on practical use of the system.

These techniques are not necessarily mutually exclusive; for example, one post-event protocol

consists of a video recording of system-user interactions that is subsequently commented on by the user. Thus, an empirical element is filtered through a subjective interpretation provided by the user.

1.3 Aims

This paper describes the VISNU (Validation of Intelligent Systems aNd Usability) tool, designed specifically to assist in the development of validation and usability analyses for intelligent systems. The most important features of VISNU are as follows:

- It integrates different methods and approaches for evaluating intelligent systems in one product.
- It includes novel aspects such as the use of artificial intelligence techniques for the interpretation of results.
- It integrates the results of different analysis methods so as to obtain information that could not be obtained by results interpreted in isolation.

Verification aspects have been excluded simply to ensure that the tool can be applied to as many systems as possible, regardless of the knowledge representation paradigm used.

The paper is laid out as follows: section 2 describes the VISNU architecture and modules; section 3 shows some examples of the application of VISNU; and finally, sections 4 and 5, respectively, contain discussion and conclusions.

2. METHODS

VISNU's novel contribution lies in its integration of several evaluation techniques in a single tool and the possibilities it offers for the combined functioning of some of these techniques. Table 1 provides a summary of the different techniques implemented in VISNU, to be commented in more detail below.

The validation module is divided into three main parts: (1) planning: that establishes the main validation strategies, (2) application: that calculates a series of quantitative measures (pair measures, group measures and agreement ratios) for analysing the intelligent system results and (3) interpretation: that tries to elucidate whether the intelligent system is really behaving as an expert within its field of application.

As for usability three kinds of techniques are considered: (1) heuristic techniques: such as the creation of GOMS [17] (Goals, Operators, Methods and Selection rules) models or ergonomic checklists. (2) subjective measures: for obtaining the opinions of users, in the form of closed questionnaires that can be analysed using MAUT [18] (Multi-Attribute Utility Theory) or AHP [19] (Analytic Hierarchy Process). And, (3) empirical techniques: such as statistical analyses for log files, identification of hierarchical tasks from log files and the possibility for instantiating GOMS models from logs and comparing predictions *a priori* with results *a posteriori*.

TABLE 1. EVALUATION TOOLS INTEGRATED IN VISNU

Validation	Planning	• Planning of validation process
	Pair measures	• Agreement index • Within-one agreem. index • Kappa • Weighted kappa • Spearman's rho • Kendall's tau and tau b • Goodman-Kruskal gamma
	Group measures	• Williams index • Hierarchical cluster • Multidimensional scaling
	Ratios	• True/false positive/negative ratios • ROC curves • Jaccard coefficient
	Interpre-tation	• Heuristic interpretation of validation results
Usability	Heuristic	• GOMS models • Ergonomic checklists
	Subjective	• MAUT questionnaires • AHP questionnaires
	Empirical	• Log analysis • Task analysis • GOMS-log integration

2.1 VISNU architecture

The architecture of VISNU is based on object-oriented programming using design patterns [20]. The inclusion of these patterns makes the tool more flexible and extendable to future modifications, so the inclusion of new modules or the modification of existing modules does not affect the other modules in the system. The VISNU project is based on the following four clearly differentiated modules (Fig. 1):

- *gui*: this includes the code for implementing the main graphical user interface that supports the interfaces for the different modules.
- *util*: this contains the helper software for the other modules.
- *validation*: this includes the modules designed for carrying out the validation processes, among which the most important are: (1) *gui*: the validation module interface, (2) *planning*: the planning system, (3) *measures*: quantitative validation measures, and (4) *interpretation*: the interpretation system.
- *usability*: this includes the modules designed for carrying out the usability analysis proce-

dures, among which the most important are: (1) *maut*: that implements questionnaires that can be analysed using MAUT, (2) *ahp*: that implements questionnaires that can be analysed using AHP, (3) *logging*: that implements the GOMS analysis, log analysis and the GOMS-log integration.

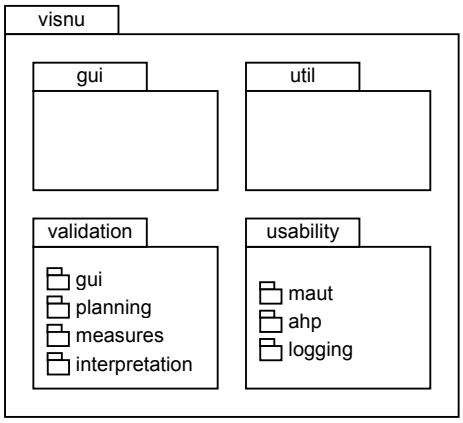

Fig. 1. VISNU modules.

The following sections will describe the validation and usability modules in more detail.

2.2 Validation module

The validation module in VISNU was developed on the basis of previous work performed by the authors on the validation tool SHIVA (System for Heuristic and Integrated Validation) [21]. This tool was designed according to a methodology that divided the process into three phases, namely, planning, application, and interpretation.

2.3 Planning

With a view to determining the most suitable validation strategies, the planning phase involves an analysis of the system characteristics, the application domain and the development phase.

Table 2 shows the criteria that are analysed in the validation planning process. For example, if the outputs of the system follow an ordinal scale, the most suitable approach is to weight the discrepancies according to importance (for example, in the symbolic processing of a given variable, a discrepancy between the categories 'very high' and 'slightly high' is not quite the same as between the categories 'very high' and 'very low'). Weighted kappa or the within-one agreement index are highly appropriate measures for taking discrepancies into account. Further details of the planning module are to be found in [21].

2.4 Application

The application phase applies the strategies identified in the planning phase by making quantitative measurements using test data. The procedure for calculating the different quantitative measures is depicted in Fig. 2.

Table 2. Criteria to be analysed in the planning phase

Subject	*Criteria to be analysed*
Application domain	• Critical domains • Validation criteria • End-user profile
System	• Division in sub-systems • Uncertainty management • Type of output variables • Type of problem in hand • Relationship with the environment
Developmen t phase	• Initial phases • Intermediate phases • Final phases

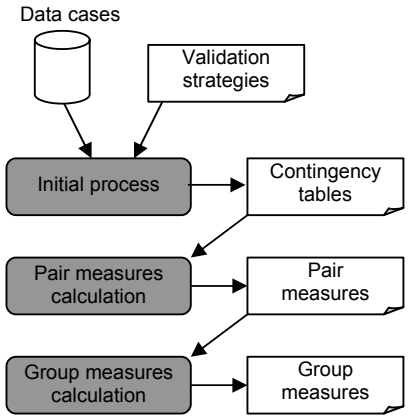

Fig. 2. Procedure for obtaining quantitative validation measures.

The first step is to analyse the existing test cases according to the validation strategies identified in the planning phase. This initial analysis permits us to construct contingency tables that correlate the interpretations of each of the possible pairs that can be formed between the experts that participate in the validation process (including the intelligent system).

Contingency tables will serve as the basis for the construction of pair measures, such as kappa or the agreement index, that provide an index that quantifies coincidences between the interpretations of two experts. Fig. 3 illustrates a contingency table and the pair tests obtained from it.

These pair measures can be used as input for the calculation of group measures, such as cluster analysis or MDS [22] (Multi-Dimensional Scaling) the objective of which is to analyse together the interpretations of the experts and to endeavour to find representation structures that permit an easier interpretation within the context of the validation. In Fig. 4 we can see a bubble graph that integrates clustering information with MDS information.

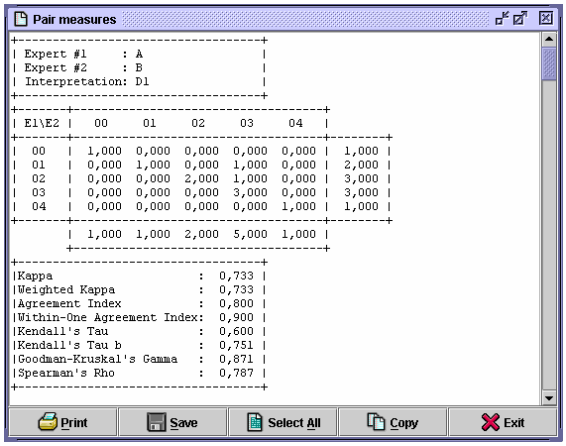

Fig. 3. Contingency table and pair tests.

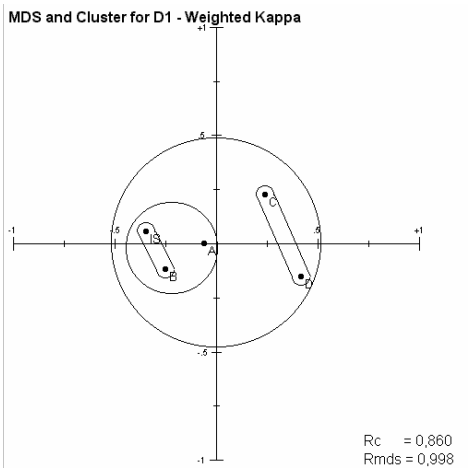

Fig. 4. Bubble graph integrating clusters and MDS information.

The advantage of VISNU is that it integrates all the validation techniques in a single module of the tool and the overall process can, therefore, be easily performed by the user. A description of these measurements can be found in [21].

2.5 Interpretation

The final phase, interpretation, involves an analysis of the results obtained in the application phase, with a view to testing whether the intelligent system genuinely behaves as yet another expert in the domain.

To facilitate the implementation of the interpretation phase it was decided to develop an expert system that would analyse the statistical results obtained in the application phase and draw conclusions on the performance of the intelligent system being validated.

The expert interpretation system is composed of two modules: an algorithmic module, based on the unprocessed data for the statistical measures, which produces output in the form of high-level information; and a heuristic module that processes this high-level information to obtain the final interpretation results [23]. The fact that the algorithmic module filters and processes the basic data to convert them into high-level information permits the rules of the heuristic module to be defined with an economy of expression.

2.6 Usability module

The VISNU usability module is designed to provide support for the different usability analysis techniques - heuristic, subjective and empirical (see Table 1). The following sections will briefly describe the different implemented techniques.

2.7 MAUT analysis

MAUT (Multi-Attribute Utility Theory) analysis is a formal subjective multi-criterion analysis technique, employed in usability environments to assess the utility of systems or alternatives that have more than one evaluable attribute [18]. The procedure for a MAUT analysis is as follows: (1) specification of the evaluation criteria and attributes; (2) weighting of these criteria and attributes according to their relative importance on a subjective manner, which leads to a subjective interpretation through an objective methodology; (3) testing how the system complies with each of the defined attributes; (4) creation of utility functions that will convert the above scores into utility measures; (5) integration of the utility values obtained for each attribute into a single measure; and (6) sensitivity analysis.

MAUT is very suitable for validating closed questionnaires (for which responses are restricted to a closed set of options). In our case the questions are used to obtain values for the attributes in our MAUT tree. Fig. 5 shows a hierarchical tree established to calculate the overall utility of a computer system [4]. It can be observed how the different attributes of the tree have been weighted according to their relative importance.

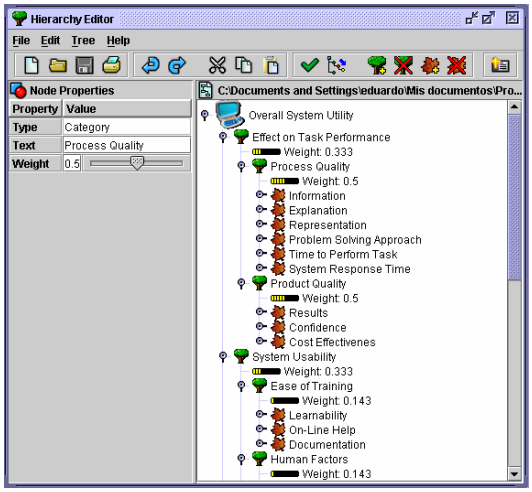

Fig. 5. MAUT criteria and attributes tree.

The questionnaire is developed by associating questions to attributes in the hierarchical MAUT tree. Because the questions do not need to have equivalent importance levels they can be weighted accordingly. The results of the evaluation of the responses to the questionnaire will give us an overall utility measure for the system being evaluated. The MAUT analysis can also be used to evaluate ergonomic checklists developed to perform a heuristic analysis of usability.

2.7 AHP analysis

One of the drawbacks to a MAUT analysis is that suitable utility functions need to be established for each study. To avoid the need to define such functions, another multi-criterion method can be used, namely, the AHP (Analytic Hierarchy Process). Developed by Saaty [19], AHP has an additional advantage over MAUT in that it permits a formal treatment of the inconsistencies that may appear in the analysis.

The drawback to AHP is that it is a comparative analysis, in other words, it is unable to reflect the utility of a single system in isolation, merely the utility of one system compared with an alternative system. AHP has other disadvantages, such as controversial ratio scales used to make the pair comparisons, or the rank reversal problem, which basically means that the AHP results may change if the number of alternatives changes.

The AHP module of VISNU can read criteria trees created for the MAUT questionnaires (Fig. 5) what allows to analyse the same problem using different tests (AHP or MAUT).

2.8 Log analysis

As we have seen in previous sections, one of the most common ways of measuring how a system is used is through an analysis of log files. In other words, the system non-intrusively records its interactions with the end-user, and these interactions are subsequently analysed to identify possible usability problems. The main drawback with the log method is that the data in the log files are generally low-level and lack context, which makes it difficult to identify the aims of the user when a log event was generated.

The log tool included in VISNU analyses log events and carries out an analysis of tasks, in other words, it identifies initial and final instants of the different tasks, errors and different device (mouse, keyboard, etc.) inputs that occur during the execution of a task. This task analysis permits the following information to be obtained:

- A statistical analysis of all the tasks executed including mean duration in time, number of instances, number of errors, etc. (Fig. 6).
- A hierarchical ordering of the tasks, with statistics about their composition (Fig. 7).
- The instantiation of a pre-existing GOMS model (described further in the next section).

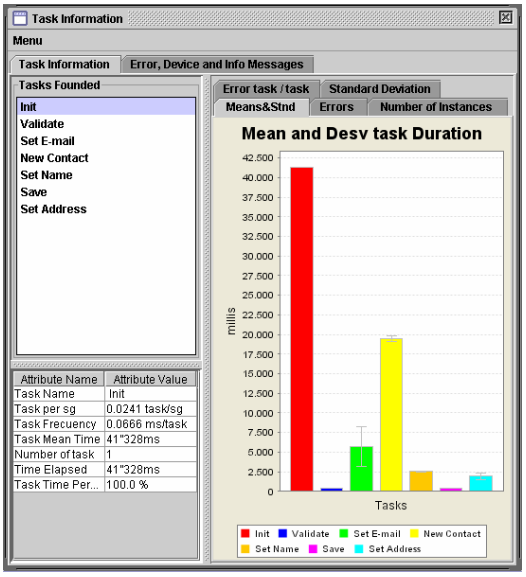

Fig. 6. Statistics for different tasks identified from the log file.

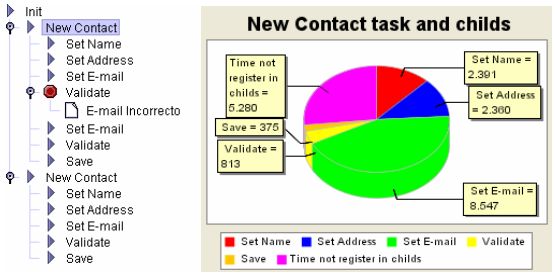

Fig. 7. Hierarchical structure for tasks identified from the log file and statistics about their composition.

2.9 GOMS-log integration

One of the novel aspects of VISNU is that it can not only make a GOMS-like heuristic analysis of usability, but can also integrate this information with the results of the empirical analysis performed by the log files. A GOMS analysis [17] is a formal analytical method for describing human-computer interaction in terms of Goals, Operators, Methods and Selection rules. The main advantage of a GOMS model is that it predicts times or sequences for the execution of commands even before the system is developed. For example, Fig. 8 shows a simple GOMS analysis for the tasks necessary to add a new contact to an agenda.

One of the main pitfalls of GOMS is that although it can be useful for the prediction of the normal user's behaviour, abnormal behaviour is not considered. Another pitfall is that GOMS models are useful for those occasions where one

wishes to check minimization of the time needed to do a task, but less appropriate to check what tasks to do in the first place, how an application copes with different human behaviours, and how well the system is structured. Having that in mind GOMS can be considered as a valuable first step in the usability analysis.

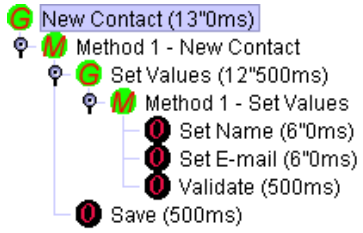

Fig. 8. GOMS model for creating a new contact.

VISNU not only permits to define GOMS models, it also permits such GOMS models to be instantiated with data from the log files. To do this, the log files need to be able to identify different tasks which must be allocated to different nodes in the GOMS tree (although this allocation does not need to be exhaustive, a more complete allocation will ensure a more accurate instantiation).

An example of an instantiation of the GOMS tree of Fig. 8 is depicted in Fig. 9. In this figure we can compare time predictions made a priori by the GOMS tree with real a posteriori results obtained in the actual use of the system. More information about GOMS and logging integration can be found in [24].

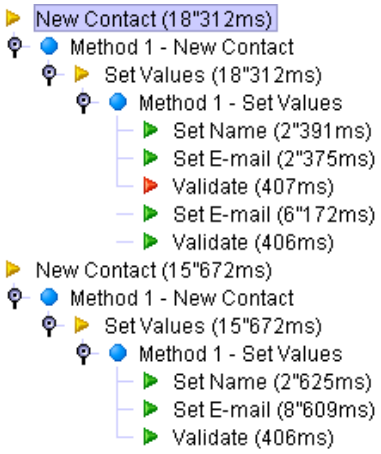

Fig. 9. GOMS model instantiated using a log file.

2.10 Module integration

The design aim for VISNU is a structure based on modules that are cohesive (abstractions that are logically related are grouped together) and loosely-coupled (dependence between modules is minimised). Fragmentation of a program in terms of individual components reduces complexity; it also creates a series of well-defined and well-documented frontiers within the program, which facilitates use and comprehension. Once the modules have been developed and tested separately, they are combined in a single application.

Another aim of the VISNU developers has been to make this tool available over the Internet by applying a rich-client philosophy. This philosophy has been adopted in view of the fact that web pages do not have the complexity necessary for the different modules of the application. Moreover, as the system is written in the Java programming language, it can be deployed via the Java Web Start platform, thus guaranteeing execution in any platform for which a Java virtual machine exists.

3. APPLICATION EXAMPLES OF VISNU

Both the VISNU tool and its antecedent SHIVA have been used for validation and usability analysis for a number of intelligent systems developed in the Laboratory for Research and Development in Artificial Intelligence (LIDIA) of the University of A Coruña, Spain. These systems are: PATRICIA [25], NST-EXPERT [26], CAFE [27], MIDAS [28] and SAMOA [29].

PATRICIA is an intelligent monitoring system for patients in Intensive Care Units. Given the critical nature of the domain, validation was performed against 6 human experts using group measures. Space does not permit a detailed description of the results of the validation for each of the PATRICIA modules. However, what we can say is that the results were more than satisfactory for all modules, and that the validation process did permit to explain the discrepancies identified.

For example, in PATRICIA the result of the module for analysing the acid-base balance was used to establish the ventilatory therapy of the patient. In the validation, however, it was discovered that although PATRICIA did not agree with the experts in the interpretation of the acid-base balance, the system did coincide in the therapy based on this interpretation. This fact can be seen in Fig. 10, in the left the results of PATRICIA (G) showed that is clearly outside the main cluster of experts, but in the right PATRICIA is very near to the origin of co-ordinates meaning that is the expert whose interpretations are closest to the consensus.

This apparent contradiction was resolved by analysing the use of context (diseases, medication, etc.) in the evaluation of the patient. PATRICIA applied the context at the moment of analysing the diagnosis whereas the human

experts used the context, not when indicating the diagnosis but when deciding the therapy. This example shows how an experienced intelligent system evaluator could take benefits from using VISNU. Complete validation results for PATRICIA can be consulted in [25].

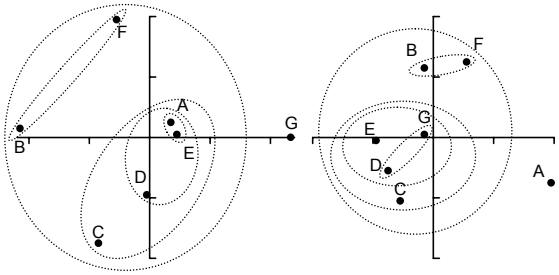

Fig. 10. Bubble graph showing the results of MDS and cluster analysis for acid-base balance interpretation (left) and ventilatory therapy (right) between human experts (A-F) and PATRICIA (G)

Other example of the application of VISNU is the SAMOA project, an intelligent monitoring system for patients with the Sleep Apnea Syndrome (SAS). In the near future, this system will be implemented as an element in daily clinical practice in the Sleep Unit of the Juan Canalejo Hospital of A Coruña. Given that the system will be used by staff with little computer knowledge, a usability analysis is needed.

The first part of the usability analysis was a heuristic evaluation based on an ergonomic checklist taken from the literature, but adapted to the particular features of our system. This ergonomic checklist was implemented using the MAUT module of VISNU. In this analysis two kinds of problems were detected related to the help system and the methods for preventing and diagnosing errors. The remaining categories obtained more than 65% of positive responses.

The second part of the analysis was carried out by issuing users with a questionnaire designed to evaluate performance aspects of the system, basically by indicating strengths and weaknesses as well as possible improvements to the system. The questionnaire had 101 questions organised into three main categories, as follows: task performance, system usability and system fit. The results of the questions, evaluated using the MAUT analysis, showed an average score of 4.08 (of the maximum of 5). The poorest results were in the areas of flexibility, user control, on-line help and documentation, and system adaptation to the user profile.

Finally, a preliminary empirical evaluation corroborated the above results in regard to deficiencies in the help function, given that the user has not used this function. A more complete description of the usability analysis of SAMOA can be consulted in [29].

As a final comment we can say that, in validation and analysing the usability of the previously mentioned systems, the VISNU tool has demonstrated to be both useful and accurate. However, as far as the own VISNU usability is concerned, only its actual use will conclusively demonstrate the validity of the followed approach. It is expected that, when the tool will be freely available in Internet, there could be obtained new experiences from other research groups.

4. DISCUSSION

Evaluation is a crucial phase within the development cycle for any computerised system. This is even truer of intelligent systems that model human expert knowledge. In view of this fact, a great deal of effort has been invested in automating the different evaluation phases. Nonetheless, in our opinion the success of these tools can be considered relative, due to the fact that no single ideal method or tool exists that is capable of implementing the different evaluation phases. What does exist is a range of methods that are particularly indicated to evaluate specific aspects of a system. Combined use of these methods may provide the desired results.

This was precisely the philosophy underlying the development of VISNU, which integrates different evaluation techniques in a single tool and thereby provides the following benefits:

- All the evaluation tools are accessible through the same interface and are distributed together. The consistency of the interface for the different modules (icons, structure, etc.) has been maintained, thereby facilitating the learning process.
- Integration of the different tools in a single system means that it is a simple matter to use the outputs of one module as the inputs for another; moreover, this can be done automatically. Pair measures, for example, can be calculated automatically when a group measure such as cluster analysis is selected.
- In some cases it is even possible to integrate the results for the different methods in a single structure; for example, bubble graphs integrate the cluster and MDS results; and GOMS models can be instantiated from log files.
- Some of the methods – such as the group and pair measures for validation – include facilities for interpreting results using an expert interpretation system. The knowledge in this system is the fruit of the accumulated system validation experience of a range of knowledge engineers.

The use of VISNU in the validation and

usability analysis of real systems permits to develop a field validation of the own system. This field-testing revealed a number of interesting aspects of the system; in particular, application of our validation methods – despite the treatment of the system as a black box – not only allowed the performance of the intelligent system to be compared with that of human experts, but also permitted it to acquire new knowledge and/or refine existing knowledge.

5. CONCLUSIONS

The main aim underlying the development of VISNU was to integrate different evaluation methods in a single tool so as to benefit from the advantages of executing various evaluation methods together.

Validation tools can be applied to any intelligent system, given that they are independent of the underlying architecture of the system. Usability analysis tools can be applied to any computerised system since they involve no specific premises.

It is also important to point out that the validation and usability analysis phases should be integrated in a natural way in the software development process. Intelligent systems are software products, and so the experience acquired by software engineers can also be applied to knowledge engineering. Nonetheless, the distinctive features of these systems and of their application domains would indicate that in terms of development and evaluation methodologies, these systems differ fundamentally from each other within the software engineering field, and for that reason specific evaluation techniques are required.

REFERENCES

[1] Mosqueira-Rey, E., Moret-Bonillo, V., "A Computer Program to Assist Developers in Validation and Usability Analysis of Intelligent Systems," *IPSI BgD Proc.*, VIP Scientific Forum of the IPSI Montenegro Conf., 2004.

[2] Adrion, W.R., Branstad, M.A., Cherniavsky, J., "Validation, verification and testing of computer software," Computing Surveys, vol. 14, no. 2, 1982, pp. 159-192.

[3] Hoppe, T., Meseguer, P. "VVT Terminology: A Proposal," IEEE Expert, vol. 8, no. 3, 1993, pp. 48-55.

[4] Adelman, L., Riedel, S.L., "Handbook for Evaluating Knowledge-Based Systems," *Kluwer Academic Publishers*, Boston, 1997.

[5] Juristo, N., Morant, J.L., "Common Framework for the Evaluation Process of KBS and Conventional Software," Knowledge-Based Systems, vol. 11, 1998, pp. 145-159.

[6] Boehm, B.W. "Software Engineering Economics," *Prentice-Hall*, Englewood Cliffs, NJ, 1981.

[7] O'Keefe, R.M., Balci, O., Smith, E.P. "Validating Expert System Performance," IEEE Expert, vol. 2, no. 4, 1987, pp. 81-89.

[8] O'Keefe, R.M., O'Leary, D.E., "Expert System Verification and Validation: a Survey and Tutorial," Artificial Intelligence Review, vol. 7, no. 1, 1993, pp 3-42.

[9] Gupta, U.G. (ed.) "Validating and Verifying Knowledge-Based Systems," *IEEE Computer Society Press*, Los Alamitos, California, 1991.

[10] Murrel S., Plant R.T. "A survey of tools for the validation and verification of knowledge-based systems: 1985-1995," Decision Support Systems, vol. 21, 1997, pp. 307-323.

[11] Swets, J.A. "Measuring the accuracy of diagnostic systems," Science, vol. 240, 1988, pp. 1285-1293.

[12] Sackman, H. "Delphi Assessment: Expert Opinion, Forecasting and Group Process," *The Rand Corporation*, Santa Monica, CA, 1974.

[13] Nielsen, J. "Usability Engineering," *Morgan Kaufmann*, San Francisco, 1994

[14] Nielsen, J., Mack, R.L., (eds.) "Usability Inspection Methods," *John Wiley & Sons*, New York, NY, 1994.

[15] Preece, J., "A Guide to Usability: Human Factors in Computing," *Addison-Wesley*, Reading, MA, 1993.

[16] Ivory, M., Hearst, M. "The State of the Art in Automated Usability Evaluation of User Interfaces," ACM Computing Surveys, vol. 33, no. 4, 2001, pp. 173-197.

[17] Card, S.K., Moran, T.P., Newell, A., "The Psychology of Human-Computer Interaction," *Lawrence Erlbaum Associated*, Hillsdale, NJ, 1983.

[18] Winterfeld, D., Edwards, W., "Decision Analysis and Behavioral Research," *Cambridge University Press*, Cambridge, England, 1983.

[19] Saaty, T.L. "The Analytic Hierarchy Process," *McGraw-Hill*, New York, 1980.

[20] Mosqueira-Rey, E., de la Rocha, J.G.F.G., Moret-Bonillo, V. "Design of a Validation Tool Based on Design Patterns for Intelligent Systems," Lecture Notes in Computer Science, vol. 2774, 2003, pp. 1365-1372.

[21] Mosqueira-Rey, E., Moret-Bonillo, V., "Validation of Intelligent Systems: A Critical Study and a Tool," Expert Systems with Applications, vol. 18, no. 1, 2000, pp. 1-16

[22] Jobson, J.D. "Applied Multivariate Data Analysis," *Springer-Verlag*, New York, 1992.

[23] Mosqueira-Rey, E., Moret-Bonillo, V. "Intelligent Interpretation of Validation Data," Expert Systems with Applications, vol 23, no. 3, 2002, pp. 189-205.

[24] Mosqueira-Rey, E., Rivela Carballal, J., Moret-Bonillo, V. "Integrating GOMS Models and Logging Methods into a Single Tool for Evaluating the Usability of Intelligent Systems," Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, 2004, pp. 5142-5147.

[25] Moret-Bonillo, V., Mosqueira-Rey, E., Alonso-Betanzos, A. "Information Analysis and Validation of Intelligent Monitoring Systems in Intensive Care Units," IEEE Transactions on Information Technology in Biomedicine, vol. 1, no. 2, 1997, pp. 87-99.

[26] Alonso-Betanzos, A., Mosqueira-Rey, E., Moret-Bonillo, V., Baldonedo, B. "Applying Statistical, Uncertainty-Based and Connectionist Approaches to the Prediction of Fetal Outcome: A Comparative Study," Artificial Intelligence in Medicine, vol. 17, no. 1, pp. 37-57, 1999.

[27] Guijarro-Berdiñas, B., Alonso-Betanzos, A., "Intelligent Analysis and Pattern Recognition in Carditocographic Signals Using a Tightly Coupled Hybrid System," Artificial Intelligence, vol. 136, no. 1, 2002, pp. 1-27.

[28] Cabrero-Canosa, M., et al. "A Methodologycal Approach to Validate the Intelligent Monitoring System 'Midas'," Int. Conf. Neural Networks and Expert Systems in Medicine and HealthCare, 2001, vol. 1, pp. 245-253.

[29] Mosqueira-Rey, E., Cabrero-Canosa, M., Hernández-Pereira, E., Moret-Bonillo, V. "Usability Analysis of an Intelligent Monitoring System," Proc. 2nd European Med. & Biol. Eng. Conf. (EMBEC'02), vol. I, 2002, pp. 758-759.

E. Mosqueira-Rey received the degree in computer science in 1994 and the PhD degree in computer science in 1998, both from the University of A Coruña, Spain. He is currently associate professor in the Department of Computer Science, University of A Coruña. He is a member of the IASTED Technical Committee on Art. Intelligence & Expert Systems.

V. Moret-Bonillo received the degree in physical chemistry in 1984, and the PhD degree in physics in 1988, both from the University of Santiago de Compostela, Spain. From 1988 through 1990 he was a postdoctoral fellow in the Medical College of Georgia at Augusta, GA. He is currently associate professor in the Department of Computer Science, University of A Coruña where he leads a research group awarded as 'group of excellence' by the regional government. He is a member of various scientific societies: IEEE, ACM, etc.