User Interface Specification Issues for Computerized Educational Systems

Andrina Granić and Vlado Glavinić*

Faculty of Natural Sciences, Mathematics and Education, University of Split, Croatia
*Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

Aside from the great progress in the field of human-computer interaction, particularly in the last years the need for the development of adaptive user interfaces in several application areas has been recognized. Our research is concentrated on intelligent tutoring systems, a generation of computerized educational systems that attempt to mimic the capabilities of human tutor. In order to support the above reasoning, in this paper we elaborate on the case study of *AKBB*, a program intended for the development of an arbitrary domain knowledge base and of its user interface. In order to specify its design and behaviour, User Action Notation is advocated as a suitable method.

Keywords: computerized educational systems, intelligent tutoring systems, Adaptive Knowledge Base Builder, adaptive user interfaces, user interface specification.

1. Introduction

The goals of human-computer interaction are to provide functional, usable and well designed computer system interfaces, thus achieving their transparency and enabling end users to fully concentrate on the work. This is the main reason why in designing human-computer communication the specification of a *look and feel* of the proposed interface, i.e. its design and essential behaviour, is quite important. The increasing desire for usable, user-centered computer systems influenced the development of more appropriate, adaptive communication between a human user and the system. Consequently, adaptive human-computer interaction constitutes a promising solution to the provision of usable user interfaces.

Rapid progress of computer technology made computer a strong tool in education altogether, resulting in the development of a generation of computerized educational systems called intelligent tutoring systems, which could be considered as emulators of human teachers in the process of learning and teaching. Since users can interpret these systems as user interfaces to some particular domain knowledge, the degree of their effectiveness and efficiency should inevitably depend on systems' usable design brought to them.

Our research is concentrated on usable, user-centered, adaptive user interfaces for intelligent tutoring systems. The paper elaborates on the user interface specification issues for *Adaptive Knowledge Base Builder, AKBB*, an arbitrary domain knowledge generator with adaptive interface, discussing the possibilities of determining the respective functionality at the user-machine interaction level, along with aspects of its adaptivity. The User Action Notation is advocated as a suitable one for *AKBB* user interface specification.

2. User Interfaces

Recent studies show that human-computer interaction as well as user interfaces, broadly defined as the two-way communication channel between the human and the functional elements of the machine, are high on the list of topics with the greatest "knowledge gap", the topic importance mostly exceeding current knowledge [15]. The design and implementation of human-computer interfaces is inherently difficult and time consuming, e.g. [19], and additional difficulties arise in the case of interfaces that are to tailor a system's interactive behaviour according to the

requirements of an interaction, thus creating the need for adaptive human-computer interaction, e.g. [12].

2.1. Adaptive User Interfaces

It has been argued that adaptive user interfaces provide the only way to achieve certain usability goals, such as meeting the users' changing knowledge and behaviour over time [3]. Within this context a user interface is called intelligent in the degree it adapts itself [13] and makes these communication decisions dynamically, at runtime [*ibid.*], [20]. Intelligent user interfaces facilitate a more "natural" user-computer interaction, attempting to imitate human-human communication and constitute a major direction in current human-computer interaction research, e.g. [16]. Several efforts towards the development of adaptive user interfaces have been reported in the literature since the early eighties when they appeared, resulting in a number of prototype systems in several application domains like intelligent help, (intelligent) tutoring, information filtering or intelligent multimedia systems [12], [14].

2.2. User Interface Specification

No matter how usable an interface design is, without a complete, thorough and understandable specification, misunderstandings can arise and unusable or incomplete user interfaces will result. Considering existing user interface specification methods it should be noted that presently there is a number of them already developed which are in a greater or lesser degree suitable for the purposes of interface definition.

Grammar or diagram approaches are suited for some of the interaction styles like menus, commands or form fill-ins; however, they are clumsy for direct manipulation interface definition, because they cannot conveniently cope with the variety of permissible actions and visual feedback the system provides [24]. In addition, direct manipulation interfaces depend heavily on the context to determine the meaning of the input. Thus alternative methods are needed which could handle the rich world of direct manipulation interfaces (pointing, dragging, clicking). Being a high-level notation to facilitate capturing the interaction design [6], [11], [19], the *User Action Notation*, UAN, is a representative of such methods.

3. User Interfaces for Computerized Educational Systems

In the field of education computer technology has great impact and influence in three main aspects: computer as subject of teaching, computer as a tool for supporting the teaching process and finally computer as the teacher itself [8], [23].

3.1. Intelligent Tutoring Systems and Authoring Shells

Contemporary efforts in computer-supported learning and teaching have introduced tutoring systems with a certain level of intelligence. *Intelligent tutoring systems*, ITSs, are the generation of computerized educational systems, which attempt to mimic the capabilities of human tutors [8]. They are intended to improve the process of learning and teaching by adjusting the contents and the way of domain knowledge perception depending on students' learning capabilities [4]. As the need to cover a variety of different domains has arisen since, instead of having a number of specialized intelligent tutoring systems, ITS generators were developed, which can be "programmed" for a particular domain by modifying the domain knowledge. Those systems are usually denoted as *authoring shells* [1], and are still quite rare.

Because of their ability to express the cognitive model of human memory and reasoning, in some intelligent tutoring systems knowledge representation is based on *semantic networks* [21]. The basic components of semantic networks are nodes, which are used for presentation of domain knowledge objects, and links, which show relations between the objects.

As highly interactive systems, ITSs rely upon a quality user interface, which should exhibit a number of properties like efficiency, effectiveness and usability. Communication between

users and the respective ITS is inherently complex, especially when supporting student interaction because of the students' dealing with concepts — the domain knowledge — yet not understood very well [17]. Moreover, as users have various preferences, experience and knowledge, what is especially true with the diversity the student population provide, need results for supplying suitable alternative ways for different users to communicate with an ITS.

3.2. AKBB — Adaptive Knowledge Base Builder, a Domain Knowledge Generator with Adaptive Interface

The vast majority of intelligent tutoring systems, as well as authoring shells, insure reasonable designs simply by pre-defining the respective interface, providing to and expecting from users to use one static user interface, i.e. not providing interface design adaptation at all [18]. Consequently, identification of users' individual differences and also their changing knowledge and behaviour over time during the system's use, as well as the incorporation of adaptivity in the respective user interface, e.g. [9], enables the improvement of intelligent tutoring system's efficiency, effectiveness and especially usability, e.g. [10].

Our research is concentrated on the incorporation of adaptivity in the user interface of the intelligent hypermedia authoring shell *Tutor-Expert System*, TEx-Sys [25], which has already been used for some time at university and high school level education, providing the means for developing specialized intelligent tutoring systems for particular domains of education. Thus, building upon the basic functionality of TEx-Sys' Developing module/shell, we conceptualized the *Adaptive Knowledge Base Builder, AKBB*, a domain knowledge generator with adaptive interface, see Figure 1.

3.3. AKBB User Interface Specification

In this paper we consider how the appropriate specification of *Adaptive Knowledge Base Builder*'s adaptive interface provides for the system's refinement and especially for the definition of its adaptivity to different users. Because of its interface-specific symbols for actions and feedback, we deem the User Action Notation appropriate enough for specify-

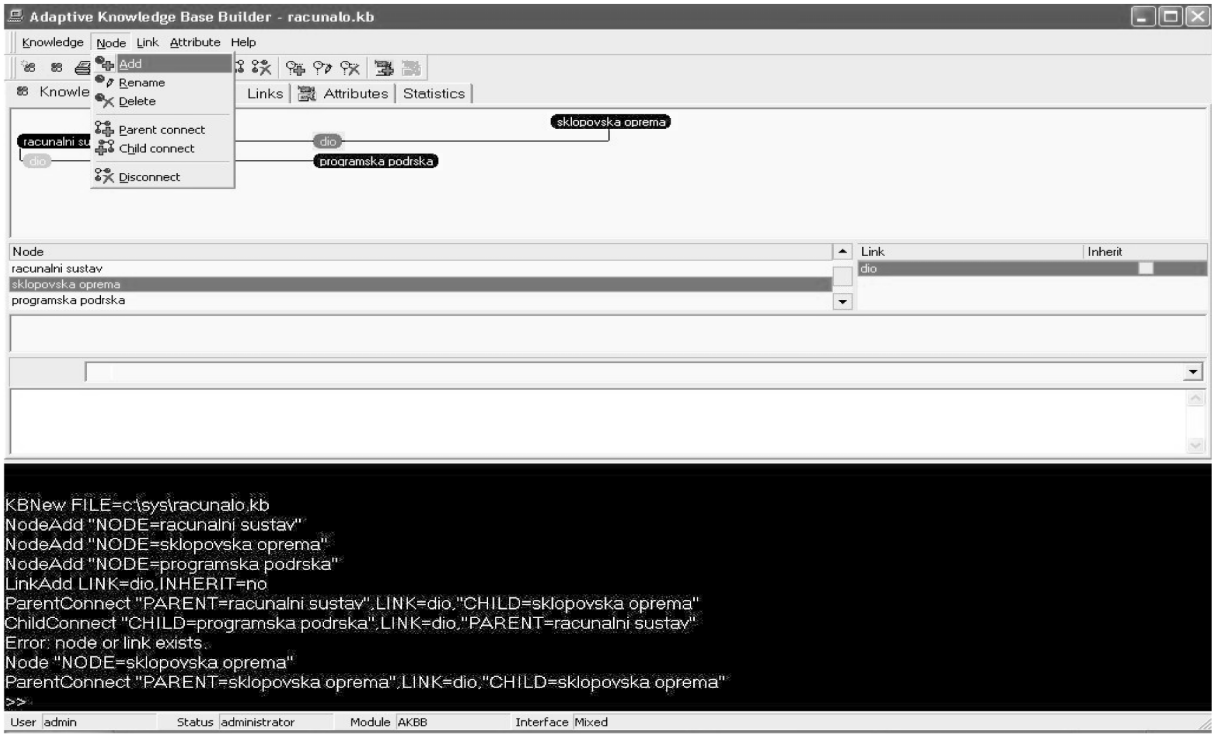

Fig. 1. Adaptive Knowledge Base Builder user interface.

ing *AKBB*'s behaviour, as well as for describing user actions.

As the principles for designing human-computer interaction prescribe, frequent users should be provided user-defined dialogue like command interfaces, while intermittent and novice users prefer system-defined dialogue like menu interfaces. Menu selection additionally reduces memory load by supporting users in making the right choice freeing them of the need to memorize specific commands. Moreover, due to their great impact on human-computer interaction, users' individual differences have to be taken into account too, e.g. [7].

Among the variety of individual cognitive characteristics, personal and/or experience profile differences that represent characteristics of individual users, we consider their *spatial ability*, *experience in command languages*, as well as *commonness of AKBB usage*. According to the users' individual differences and their changing knowledge and behaviour during the interaction, the values of these parameters are changing, enabling adequate input for the set of inferring as well as adaptive rules. Both sets provide adequate automatic adaptation of *Adaptive Knowledge Base Builder* user interface. Three different kinds of interfaces are provided: (i) a *command interface* which only enables interaction through the command line, (ii) a *graphical interface* which provides an interaction to be performed using combination of direct manipulation and menu selection, enhanced with form fill-ins and mouse right button functions and finally (iii) a *mixed interface* providing the combination of the former two.

Considering functional requirements which describe system capabilities, namely requirements which "*. . . specify the system function to be supported [and provide] the functional objectives [which] point to the behavior that is to be supported*" [5, p. 15], at the *Adaptive Knowledge Base Builder* task level the execution of just one task is supported - generation of an arbitrary domain knowledge base. The *task* level [2], or according to [22] *goal* or *external task*, is mapped into two levels:

- the *logical* level, or (*internal*) *task* [*ibid.*], which maps every task into subtasks, i.e. system logical functions; e.g. the task *Knowledge Base Generation* is mapped into logical functions *1. Create new knowledge base, 2. Open existing knowledge base, 3. Add new node to knowledge base* etc., and
- the *physical* level, or *action* [*ibid.*], which maps every task into physical actions within the interface; as more than one interface is provided, the physical level is defined for each one of them, e.g. in the graphical interface the logical function *3. Add new node to knowledge base* is mapped into physical actions: *3.1.* select an option *Node* from *main_menu*, *3.2.* select an option *Add* from *node_menu*, *3.3.* write *new_node_name* in *Node add* form fill-in and *3.4.* click <*Add*> button.

As an illustration, the mapping of the logical function *Add new node to knowledge base* into the respective physical actions for all three kinds of interfaces is presented in Table 1 and subsequently used to show one application. The function *Add new node to knowledge base* in the graphical interface can be described in prose as follows:

1. move the cursor to the *Node* alternative in the first-level selection of the *main_menu*; push and release the left mouse button;
2. pushing and releasing the button drops down the second-level selections for *Node* alternative;
3. move the cursor to the *Add* alternative in the second-level selection; push and release the left mouse button;
4. pushing and releasing the mouse button enables creation of *Node add* form fill-in;
5. write *new_node_name* of the domain knowledge in a *Node* field of a *Node add* form fill-in;
6. move the cursor to the <*Add*> button of the *Node add* form fill-in; push and release the left mouse button;
7. pushing and releasing the mouse button enables creation of the new node named *new_node_name* in the Knowledge panel;
8. move the cursor to the <*Cancel*> button of the *Node add* form fill-in; push and release the left mouse button key;

LOGICAL LEVEL	PHYSICAL LEVEL		
	graphical interface	**mixed interface**	**command interface**
Add new node to knowledge base	select an option *Node* from *main_menu* select an option *Add* from *node_menu* write *new_node_name* in *Node add* form fill-in click <*Add*> in *Node add* form fill-in click <*Cancel*> in *Node add* form fill-in	the combination of both graphical and command interfaces	write **NodeAdd** or **NA** push <*Enter*> write *new_node_name* push <*Enter*>

Table 1. Illustration of functional requirements for *AKBB*.

9. pushing and releasing the mouse button closes the *Node add* form fill-in.

It can be noted that the above description consists of user actions along with interface feedback information, which is the interface response to user actions. Since user actions are interleaved with interface feedback, there is no line-by-line association of feedback with the corresponding user actions [11]. The user action portion along with the interface feedback as described with an UAN specification offers a more precise correspondence between feedback and separate user actions in the sequence, as shown in Table 2. Consequently, the main advantage of such notation is its user-centered orientation, the possibility to write down the actions that the user will perform with the *Adaptive Knowledge Base Builder* and respective system responses, but from the user's point of view. The use of interface specific symbols and the incorporation of task and context considerations in the user interface makes a suitable apparatus for *AKBB* interface specification.

4. Conclusion

Present day interactive applications make it an ever-increasing requirement to develop user interfaces that exhibit adaptivity, leading to the still open field for research intended to improve the respective interface design. This is even more important for computerized educational systems, which are not only to support a more efficient and effective interaction, but also to

TASK: Add new node to knowledge base	
USER ACTIONS	**INTERFACE FEEDBACK**
~[*Node*_selection] M∨∧	display(*Node_menu*)
~[*Add*_selection] M∨∧	display(*Node_add* form fill-in)
keyboard"new_node_name"	
~[*Add*_button in Node_Add_form_fill-in] M∨∧	display(*new_node_name* in Knowledge_panel)
~[*Cancel*_button in Node_Add_form_fill-in] M∨∧	close(*Node_add* form fill-in)

Table 2. UAN specification of user action *Node Add* in *AKBB* graphical interface and appropriate interface feedback.

change the provided interaction styles on the basis of users' individual characteristics in order to address their changing knowledge and behaviour over time.

The above reasoning can be supported by considering the case of the *Adaptive Knowledge Base Builder, AKBB*, an arbitrary domain knowledge generator with adaptive interface, which provides the means for the development of specialized intelligent tutoring systems for particular domains of education. *Adaptive Knowledge Base Builder* is an attempt to overcome problems due to both the increasing complexity and sophistication of human-computer interaction and individual end user needs in computerized educational systems in general. In order to specify *AKBB*'s design and behaviour the User Action Notation is advocated as a suitable method because of its use of interface specific symbols and the incorporation of task and context considerations.

5. Acknowledgements

This work has been carried out within project 0036033 *Semantic Web as Information Infrastructure Enabler*, funded by the Ministry of Science and Technology of the Republic of Croatia.

References

[1] BARTON M. Authoring Shells for Intelligent Tutoring Systems. *7th World Conference on Artificial Intelligence in Education*, Washington, USA, August 16–19, 1995, http://www.pitt.edu/~al/aied/barton.html

[2] BENYON D. *Employing Intelligence at the Interface*. Handbook of UI chapter, draft 1, unpublished book chapter, 1998. www.dcs.napier.ac.uk/~dbenyon/IIT.pdf

[3] BENYON D. Intelligent Interface Technology. In Howard S, Hammond J, Lindgaard G, editors. *Human-Computer Interaction: INTERACT'97*, Chapman & Hall, pp. 678–679, 1997.

[4] BURNS H., CAPPS C. Foundations of Intelligent Tutoring Systems: An Introduction. In Polson M, Richardson J, editors. *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, 1988, pp. 1–18.

[5] CARROLL J., ROSSON M. Usability Specifications as a Tool in Iterative Development. In Harston R, editor. *Advances in Human-Computer Interaction 1*, Ablex, Northwood, NJ, 1985, pp. 1–28.

[6] CHASE J., SCHULMAN R., HARTSON R., HIX D. Development and Evaluation of a Taxonomical Model of Behavioral Representation Techniques. *CHI'94 Conf. Proceedings*, 1994, pp. 159–165.

[7] EGAN D. Individual Differences in Human-Computer Interaction. In Helander M, editor. *Handbook of Human-Computer Interaction*. Elsevier Science B.V. Publishers (North-Holland), 1988, pp. 543–568.

[8] FLEISCHMANN A. The Electronic Teacher: The Social Impact of Intelligent Tutoring Systems in Education, 2000, http://www.student.informatik.tu-darmstadt.de/~andreasf/inhalte/its.html

[9] GRANIĆ A., GLAVINIĆ V. Adaptive Intelligent Tutoring Systems in the Context of Usability Requirements. 5th *IEEE Int. Conference on Intelligent Engineering Systems INES 2001*, Helsinki, Finland, September 16–18, 2001, pp. 231–234.

[10] GRANIĆ A., GLAVINIĆ V. Interface Redesign Issues for Intelligent Tutoring Systems. 9th *International Conference on Human-Computer Interaction HCI International 2001*, Poster Sessions: Abridged Proceedings, New Orleans, Louisiana USA, August 5–10, 2001, pp. 133–135.

[11] HARTSON R., SIOCHI A., HIX D. The UAN: A User-Oriented Representation for Direct Manipulation Interface Designs. *ACM Trans on Information Systems*, Vol. 8, No. 3, 1990, pp. 181–203.

[12] Intelligent Interfaces Special Interest Group, IISIG, 1998, http://www.dcs.napier.ac.uk/~dbenyon/iisig2.html

[13] KARAGIANNIDIS C., STEPHANIDIS C. Run-Time Adaptation in Intelligent User Interfaces: Automatic, Iterative Decision Making with Feedback. 1998, http://www.dfki.de/etai/articles/stephanidis-jan-98/paper.html

[14] KOBSA A. User Modeling and User-Adapted Interaction. CHI'94 *Tutorial Notes*, 1994, http://www.acm.org/sigc|hi/c|hivas/kobsa.html

[15] LETHBRIDGE T. What Knowledge Is Important to a Software Professional? *IEEE Computer*, May, 2000, pp. 44–50.

[16] MAYBURY M. Intelligent User Interfaces: An Introduction. HCI'99 Tutorial, *8th International Conference on Human-Computer Interaction HCI'99*, Munich, Germany, August 22–27, 1999.

[17] MILLER J. The Role of Human-Computer Interaction in Intelligent Tutoring Systems. In Polson M, Richardson J, editors. *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, 1988, pp. 143–189.

[18] MURRAY T. Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art. *International Journal of Artificial Intelligence in Education*, Vol. 10, 1999, pp. 98–129.

[19] MYERS B., editor. *Languages for Developing User Interfaces*, Jones and Bartlett Publishers, Inc., 1992.

[20] NORCIO A., STANLEY J. Adaptive Human-Computer Interfaces: A Literature Survey and Perspective. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 2, 1989, pp. 399–408.

[21] NUTE D. Knowledge Representation. In Shapiro S, editor. *Encyclopedia of Artificial Intelligence*, John Wiley & Sons, Inc, 1992, pp. 743–869.

[22] PREECE J., ROGERS Y., SHARP H., BENYON D., HOLLAND S., CAREY T. *Human-Computer Interaction*, Addison-Wesley, Wokingham, England, 1994.

[23] SHERWOOD R. Models of Computer Use in School Settings. In Kinzer Ch, Sherwood R, Bransford J, editors. *Computer Strategies for Education: Foundations and Content-Area Applications*. Merrill Publishing Company, Columbus, Ohio, 1986, pp. 105–130.

[24] SHNEIDERMAN B. *Designing the User Interface. Strategies for Effective Human-Computer Interaction*, 3rd Ed., Addison-Wesley, Reading, MA, 1998.

[25] STANKOV S. *Isomorphic model of the system as the basis of teaching control principles in the intelligent tutoring system*. Ph.D. Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, 1997 (in Croatian).

Received: June, 2002
Accepted: September, 2002

Contact address:
Andrina Granić
Faculty of Natural Sciences,
Mathematics and Education
University of Split
Nikole Tesle 12, 21000 Split, Croatia
Phone: (385) 21-38 51 33/105
Fax: (385) 21-38 54 31
e-mail: andrina.granic@pmfst.hr

Vlado Glavinić
Faculty of Electrical
Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
Phone: (385) 1-612 99 55
Fax: (385) 1-612 96 53
e-mail: vlado.glavinic@fer.hr

ANDRINA GRANIĆ is a research assistant at the Faculty of Natural Sciences, Mathematics and Education, University of Split, Croatia, since 1990. She holds the Doctorate, M.Sc. and the Dipl.-Ing. degrees in Computer Science from the University of Zagreb, Croatia. Her main research interests are in the areas of computerized educational systems and human-computer interaction, including adaptive user interfaces and usability evaluation.

VLADO GLAVINIĆ is an associate professor of computer engineering at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He holds the Doctorate and M.Sc. degrees in Computer Science and the Dipl.-Ing. degree in Electrical Engineering, all from the University of Zagreb.

His main research interests are in the areas of computer networks, user interfaces and digital systems design. He has collaborated in and led a number of research and industrial R&D projects, and has authored or co-authored more than 90 research and professional papers in journals, conferences and books, along with several educational texts. He has taught in a series of successful seminars for the industry, and has also consulted both for Croatian industry and the Government. He was member of the team that developed for the Croatian Government the ICT part of the Strategy of Development *Croatia in 21st Century*.

Dr. Glavinić has been involved in the organization and has been member of program committees of a number of scientific and professional conferences at the national and international level and was Chairman of the International Program Committee of ITI'2002. He serves as an Editor for CIT — Journal of Computing and Information Technology, and is member of IEEE, IEEE Computer Society, IEEE Communication Society, as well as ACM and ACM SIGCOMM.