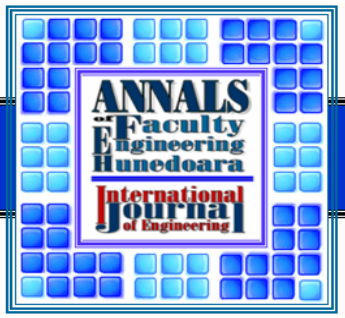
ANNALS OF FACULTY ENGINEERING HUNEDOARA
- INTERNATIONAL JOURNAL OF ENGINEERING
Tome IX (Year 2011). Fascicule 3. (ISSN 1584 - 2673)



[1.] Anca IORDAN, [2.] Manuela PĂNOIU, [3.] Ionel MUSCALAGIU, [4.] Raluca ROB

DESIGN OF AN EDUCATIONAL INFORMATICS SYSTEM FOR THE STUDY OF THE QUADRILATERAL USING UML DIAGRAMS

[1.] TECHNICAL UNIVERSITY OF TIMISOARA, ENGINEERING FACULTY OF HUNEDOARA, ROMANIA

ABSTRACT: The informative society needs important changes in educational programs. The informational techniques needs a reconsideration of the learning process, of the programs, manuals structures, a reconsideration of the methods and organization forms of the didactic activities, taking into account the computer assisted instruction and self instruction. This paper presents the necessary stages in implementing an informatics system used for the study of quadrilateral. The modelling of the system is achieved through specific UML diagrams representing the stages of analysis, design and implementation, the system thus being described in a clear and concise manner.
KEYWORDS: Educational Software, Java, Quadrilateral, Distance Education, UML

❖ INTRODUCTION

In the condition of informatics society whose principal source in the social-economic development is to produce and consumption the information, the complex and fast knowledge of the reality for rational, opportune, effective decisions is a desideratum which generate the necessity to form some superior level habituation in information manage for the whole population. The computers and their programs offer to the users powerful capabilities for the information manipulation: image and text visualize on the screen which can be manipulate later; memory storage of an important quantity of information, his accessing and selection of a part of them; possibility to realize a great volume of computation; possibility of equipment control and fast decisions; computer based training [1]. This facilities offer to the microcomputers higher educational capabilities versus other technologies used in education and provide learning controlled based on many parameters: intellectual aptitude, level of knowledge, abilities, rhythm of work.

❖ COMPUTER BASED TRAINING AS A DIDACTIC METHOD

The informatics society makes sensitive modification in education programs. In this scope, the school must prepare programmers, maintenance technicians, etc. In the same time it is necessary that the teacher make ready to use the computer in education process. These informational techniques impose to reorganize the contents of the education process, of the programs, course books and manuals, to reconsider the methods and organization forms of didactic activities, which follow to be centre on individualization of the teaching process [2]. On the other hand, programmed teaching assumes some principles which the teaching program must respect [3]:

- The small steps principle consists in progressive penetration, from simple to complex, in a subject content which logic divided in simple units series lead to minimal knowledge, which later will form an ensemble.
- The principle of personal rhythm of study regard mannerism observance and capitalization of each user of the program which will be able to make the sequences of knowledge learning or control, in a personal rhythm appropriate to his psycho-intellectual development, without time limits.
- The active participation principle, or active behaviour, regard user effort trend into selection, understanding and applying the necessary information in elaboration of a correct answer
- The principle of inverse connection, regard positive or negative inputs of user competence, refer to the success or breakdown in task performed.
- The immediate and directly control of the task work precision with the possibility to progression to the next sequence, in case of success.
- The repetition principle, based to the fact that the programs are based on return to the users initial knowledge.

The combined programming interposes the linear and branch sequence according to teaching necessities.

The expert system consists of self-teaching training programs, tutorial strategies, and the usage of natural language, mixed initiative and some complex representation of knowledge usage [4].

The computer based programmed teaching realize learning process with a inputs flow – the command, an executive controlled system, an output flux – control and a control system functions which correct measure establish.

In such a system have tree stages of teacher perceive: teaching, evaluating and the feedback loop closing, the computer being present in all of tree stages.

❖ UML

Unified Modelling Language (UML) is a standardized general-purpose modelling language in the field of software engineering [5]. UML includes a set of graphical notation techniques to create abstract models of specific systems.

The Unified Modelling Language (UML) is an open method used to specify, visualize, construct and document the artefacts of an object-oriented software-intensive system under development. UML offers a standard way to write a system's blueprints, including conceptual components such as: actors, business processes, system's components and activities, as well as concrete things such as: programming language statements, database schemas and reusable software components.

UML combines the best practice from data modelling concepts such as entity relationship diagrams, business modelling (work flow), object modelling and component modelling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has succeeded the concepts of the Booch method, the Object-modelling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modelling language. It is very important to distinguish between the UML model and the set of diagrams of a system.

UML diagrams represent two different views of a system model [6]:

- Static view: Emphasizes the static structure of the system using objects, attributes, operations and relationships. The structural view includes class diagrams and composite structure diagrams.
- Dynamic view: Emphasizes the dynamic behaviour of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams.

UML models can be exchanged among UML tools by using the XMI interchange format.

❖ DEVELOPMENT STAGES OF THE EDUCATIONAL INFORMATICS SYSTEM - SYSTEM'S ANALYSIS

Using the UML modelling language, the analysis of an informatics system consists in drawing the use case and activity diagrams [7]. The software utility ArgoUML [8] was used to construct the diagrams.

The informatics system will be described in a clear and concise manner by representation of the use-cases. Each case describes the interaction between the user and the system. The diagram defines the system's domain, allowing visualization of the size and scope of the whole developing process.

For each use-case in the diagram presented earlier an activity diagram is constructed. Each diagram will specify the processes and algorithms that are behind the use cases studied. Activity diagrams [9] are represented by nodes (with partitions and branches) or conditional blocks (with decisions). The activity diagrams are used to visualize, specify, build and document dynamic issues related to the informatics system processes. They focus on flow control seeking the transition, in a certain order, from one activity to another.

❖ SYSTEM'S DESIGNING

Conceptual modelling [10] allows the identification of the most important concepts for the system. Since classes are concepts, the following two diagrams present the classes that will be used in the project. Figure 1 presents the inheritance relationships used. It may be noted that all attributes and methods of the *Element2D* class will apply to the derived class *Patrulater2D*.

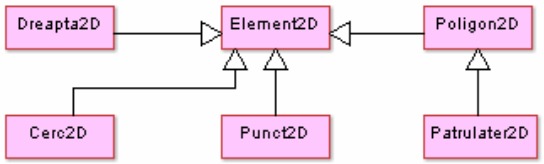


Figure 1. Class Diagram: the inheritance relationships

The composition relationships that exist between instances of the classes in the architecture are shown in figure 2. The difference of the composition relationship, with respect to aggregation, is that the instance of the whole could not exist without part objects. When looking at figure 2 one can see that a type *Patrulater2D* instance consists of four *Punct2D* type items, four *Segment2D* object type and four *Unghi2D* object type.

Both class diagrams shown contain specific classes to the application as well as existing classes and interfaces from Java.

The sequence diagram [11] is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them.

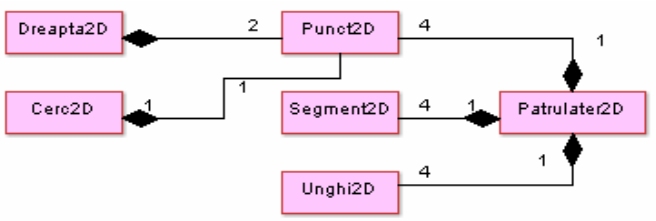


Figure 2. Class Diagram: The composition relationships

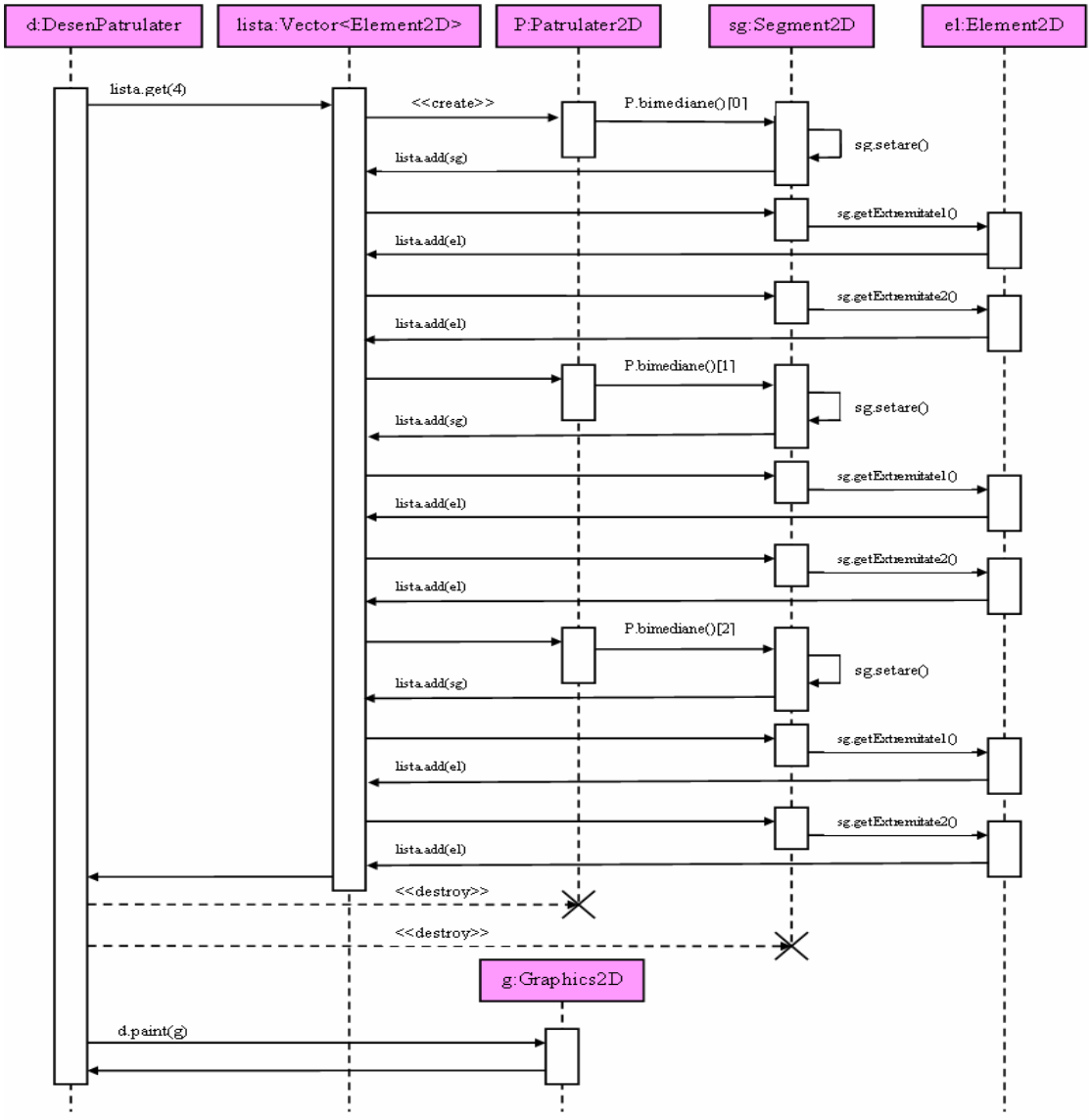


Figure 3. Sequence diagram for drawing the bimedians of a quadrilateral

The sequence diagrams for this software are made with ArgoUML-0.24. The diagram illustrate in figure 3 shows the interactions between objects, which have as purpose the drawing of the bimedians of a quadrilateral.

One can notice that there are interactions between six objects, out of which the objects of *DesenPatrulater*, *Vector<Element2D>*, *Patrulater2D*, *Segment2D* and *Element2D* type are already created, and the object *Graphics2D* type will instantiate during the interactions.

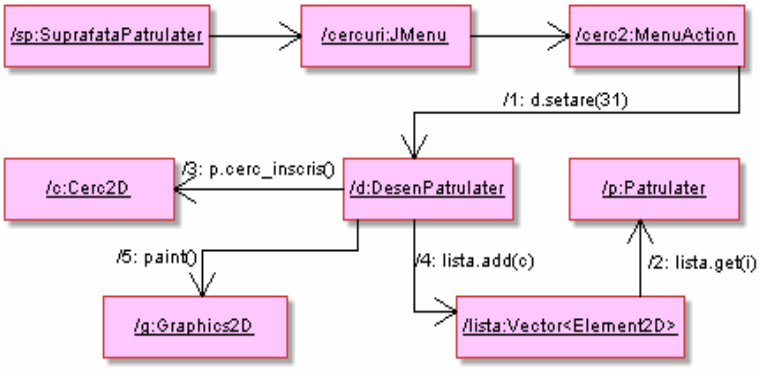


Figure 4. Collaboration diagram for drawing in the circle of a quadrilateral

Collaboration diagram, on the other hand, focus upon the relationships between the objects [10]. They are very useful for visualizing the way several objects collaborate to get a job done and for comparing a dynamic model with a static model (figure 4). Collaboration and sequence diagrams describe the same information, and can be transformed into one another without difficulty. The choice between the two depends upon what the designer wants to make visually apparent.

❖ SYSTEM'S IMPLEMENTATION

The component diagram [11] allows the visualization of the module in which the system is broken into and the dependencies between modules. The component diagram emphasis on physical software components (files, libraries, executables) and not on logic components, such as packages. The diagram in figure 5 describes the collection of components that all together provide functionality for the educational informatics system.

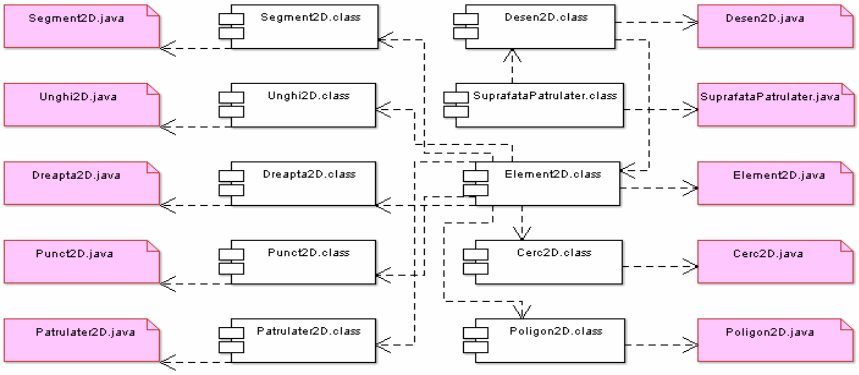


Figure 5. Component Diagram

❖ USER INTERFACE

The educational informatics system is accomplished using the Java programming language [12]. The main page of the application contains buttons for selecting the following options: Newton point and Miquel point (figure 6), Mathot point, bimedians of a quadrilateral, Newton line and Gauss line (figure 7), Aubert line for a convex quadrilateral and for a concave quadrilateral (figure 8).

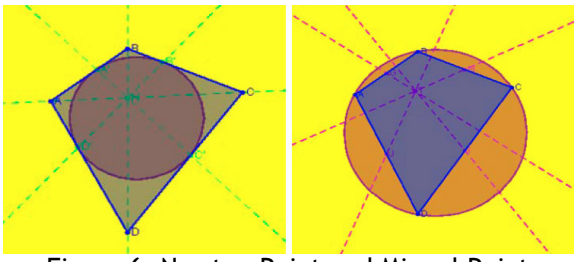


Figure 6. Newton Point and Miquel Point

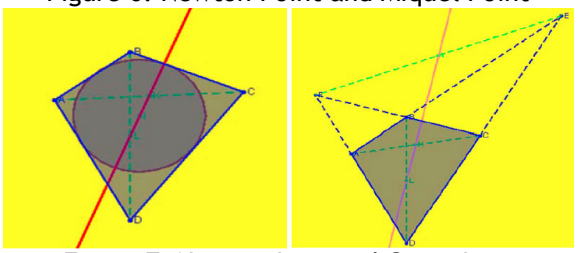


Figure 7. Newton Line and Gauss Line

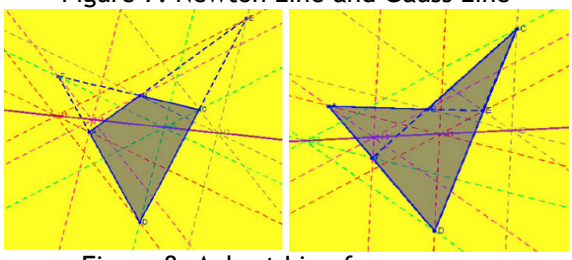


Figure 8. Aubert Line for a convex quadrilateral and for a concave quadrilateral

❖ CONCLUSIONS

Through the diagram representation all three phases: analysis, design and implementation, the educational informatics system has been described in a clear and concise manner. The use of the UML modelling language for the creation of the diagrams is characterized by rigorous syntactic, rich semantic and visual modelling support.

The diagrams were made using a new approach, multidisciplinary of the informatics application, encompassing both modern pedagogy methods and discipline-specific components. The link between teaching activities and scientific goals and objectives was established through the development of the new methods and the assimilation of new ways, capable of enhancing school performance, enabling students to acquire the knowledge and techniques required and apply them in optimum conditions.

❖ REFERENCES

[1.] A. Adăscăliţei, *Instruire asistată de calculator. Didactica informatica*, Editura Polirom, 2007
[2.] D. Glusac, D. Radosav, D. Karuovic, D. Ivin, *Pedagogical and Didactic-Methodical Aspects of E-learning*, 6th WSEAS International Conference on E-ACTIVITIES, Tenerife, Spania, 2007, pp. 67-75
[3.] I. Negret-Dobridor, I. Panişoara, *Ştiinţa învăţării. De la teorie la practică*, Editura Polirom, 2005
[4.] I. Nicola, *Tratat de pedagogie şcolară*, Editura Aramis, 2003
[5.] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1999
[6.] M. Fowler, K. Scott, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Addison Wesley, Readings MA, USA, 2000
[7.] D. Bocu, *Iniţiere în modelarea orientată obiect a sistemelor soft utilizând UML*, Editura Albastră, Cluj-Napoca, 2002
[8.] http://argouml.tigris.org
[9.] J. Odell, *Advanced Object Oriented Analysis& Design using UML*, Cambrige University Press, 1998
[10.] J. Rumbaugh, I. Jacobson, G. Booch, *The Unified Modeling Language Reference Manual*, Addison Wesley, 1999
[11.] D. Bocu, *Modelare orientată obiect cu UML*, Editura Albastră, Cluj-Napoca, 2006
[12.] S. Tănasă, C. Olaru, S. Andrei, Java de la 0 la expert, *Polirom Press*, Iasi, 2007