

# A Light Version of UML 2: Survey And Outcomes

**Stanisław Wrycza and Bartosz Marcinkowski**  
**Department of Business Informatics, University of Gdańsk,**  
**Gdańsk, Poland**

[swrycza@univ.gda.pl](mailto:swrycza@univ.gda.pl) [bartosz.marcinkowski@univ.gda.pl](mailto:bartosz.marcinkowski@univ.gda.pl)

## Abstract

UML 2 has become even more complicated and diverse set of graphical techniques than its predecessors. Therefore, system developers propose creation of its reduced, limited or minimal version called Light UML. This problem has become also the serious challenge for the UML academic teachers. The goal of this paper is the study of specifying of the UML 2.x Light version content on the basis of the questionnaire survey of opinions of 180 university students of two universities – public and private in Gdansk. After the Introduction, the methodological prerequisites of the survey are clarified. Then, the research results are presented and discussed according to seven essential UML diagrams assessment criteria, included in a questionnaire. The final UML2.x version, resulting from the accomplished survey is exposed in the last part of the paper.

**Keywords:** UML 2.x, UML Light Version, UML Teaching, Questionnaire Survey, Use Case Diagrams, Class Diagrams, Sequence Diagrams, Activity Diagrams.

## Introduction

Unified Modeling Language (UML), proposed by Booch, Jacobson and Rumbaugh (2004), has attracted the attention of both academics and practitioners of information systems analysis and design. In the last few years, increasing interest in UML stimulated spreading it across computing curricula at universities. This tendency evoked the exchange of ideas regarding the effective teaching of UML among the language trainers. Version 2.0 (OMG, 2005) and the working drafts of future UML versions (OMG, 2006) are in fact a diverse and in some parts excessive toolbox, which combined with system development process create a methodological platform for developing a working system.

Most of the UML teachers stress the question of the language complexity and variety of its modeling constructs. They consider this issue as a fundamental problem from a teaching point of view. On the basis of practical projects and teaching experiences it may be stated that only purposefully selected part of the complete UML potential is used. Moreover, a few diagrams and sets

---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org) to request redistribution permission.

of UML notions are known to form the core of a typical system model. There are versatile opinions what specific modeling notions are the most required for teaching and practical aims. Such set of UML diagram types and notions might create its minimal set or – as it is commonly called – UML Light version.

The question of the effective implementation of UML in education, in respect

of a UML Light version concept, has already been raised in different papers. Flint, Gardner, and Boughton (2004) indicate a number of problems associated with UML teaching. They accent that the use of strict subsets of UML is easier to understand than the full language notation. Burton and Bruhn (2004) generalize their experiences related to use of the UML and underline the role of CASE tools application in UML teaching. In their opinion such tools are important factors, stimulating support of the active students' involvement in teaching process as well as allowing enrichment of system specifications by using stereotypes. The concept of minimal set of UML diagrams was also proposed by DeLooze (2005). It seems that the quickness of UML upgrading and modifications, as well as potential difficulties in getting familiar with the language by novices, are underestimated. The goal of this paper is the study of specifying the UML 2.x Light version content on the basis of the questionnaire survey of the university students' opinions.

The courses of UML (2.0 and earlier versions) have been given at the University of Gdansk since 2001. The complete UML teaching approach was implemented soon after and then continuously modified and improved with each released UML version. The UML teaching process is discussed in detail in Wrycza and Marcinkowski (2005). The authors identified and analyzed several problems described in Wrycza and Marcinkowski (2006). One of the essential conclusions, being in accordance with the opinions expressed by authors cited above, is that the students are overwhelmed by the number of different UML diagrams (13 in UML 2.0), and the extensive number of modeling notions strengthened the complicated interrelationships among them. Also the survey carried out among 171 practitioners was directed at the UML version that would have a limited scope (Dobing & Parsons, 2006). The following constraints should concern such Light version:

- Light version would only consist of diagrams that are most often used in practice and would include only part of the current, detailed syntax;
- The minimal UML version should support the RUP basic disciplines, i.e. requirements specification as well as analysis and design;
- Light version should be entirely compatible with the “full” version of UML 2.x.

This concept does not limit the UML potential as the system specifications elaborated in the Light version could be subsequently extended towards the full version by the use of complete scope of UML modeling diagrams and constructs.

## Methodological Background

To solve the problem of UML Light version concept, the authors decided to carry out the questionnaire survey among two universities students. The target group encompassed 180 students within knowledge of both structured and object-oriented methodologies of systems development. All students taking part in the survey formed a competent target group, as they:

- participated in the 30 hrs lecture of UML 2.0;
- have studied the extensive UML manual entitled “UML2.0 in information systems modeling” (Wrycza, Marcinkowski, & Wyrzykowski, 2005);
- exercised the fluency in UML diagramming by solving the specified design problems using UML 2 diagrammatic notation with the support of Sparx Systems Enterprise Architect CASE tool;
- developed small UML projects in 3-4 students groups;
- had access to extensive e-learning content, supporting the course;
- in many cases the students had practical working experience as programmers or designers (in particular group leaders) .

The appropriate questionnaire containing 17 basic questions, handed to was elaborated and handed to 180 UML course students. The questions were focused around Light version concepts, reciprocal influence of structured and object-oriented approach as well as possible UML extensions. To make the proper assessment of the UML 2.x Light version the following seven crucial issues, raised in questionnaire, have been analyzed:

1. UML complexity level,
2. number of UML diagrams,
3. usefulness of the specific diagrams,
4. choice of diagrams overwhelmed with notions,
5. selection of the user-friendly UML diagrams,
6. use of the UML diagrams for the source code generating,
7. assessment of the appropriateness of the dynamics diagrams for the Light version support.

The assessment of the above problems in the synthetic opinions of interviewees is discussed in detail in the next point.

## Selected Results of the Survey

### Complexity Level

The initiating enquiry of the questionnaire regarded UML complexity (Figure 1). It's a basic question for justification the necessity for introducing UML Light version. Classifying UML 2.x as an easy or very easy technique by most of the respondents would in fact deny the concept of the Light version introduction. The students' answers, however, confirmed the authors' hypothesis – according to the students' assessment UML is most frequently classified as moderately difficult (51%), rather difficult (33%) or very difficult (7%). It means that more than 90% of respondents would welcome the more introductory, i.e. the Light version of UML.

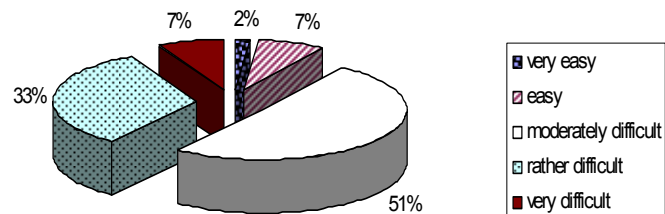


Figure 1: UML complexity level

### Diagrams Cardinality

The students exercised all 13 types of UML diagrams. The number of UML diagrams is in a natural way related to the UML complexity. Majority of interviewees (over 57%) assessed that the UML standard comprises too many types of diagrams, as shown in Figure 2. The remaining respondents accepted all types of diagrams, not assessing however the potential surplus of modelling notions number used in each type of diagram.



Figure 2: Adequacy of the number of UML diagrams

### Usefulness of the Specific Diagrams

Since only the part of the formal UML specification is used in practice, the problem of uselessness of the specific diagrams arises. The survey revealed that the future system analysts propose the following diagrams as the most useful ones (Figure 3):

- Class Diagrams (62% of accepting responses),
- Use Case Diagrams (56%),
- Activity Diagrams (26%),
- Sequence Diagrams (21%).

The investigations acknowledged commonly recognized leading role of Class Diagrams and Use Case Diagrams as the basic graphical formalisms for object-oriented modeling of the structure and dynamics information system respectively. Supplementary, Use Case Diagrams initiate incremental-iterative lifecycle in RUP and the other IS object-oriented methodologies. On the other hand, State Machine Diagrams (28%), Timing Diagrams (19%), Deployment Diagrams (13%) and Composite Structure Diagrams (12%) are recognized as the most useless diagrams. In opinion of teachers, students underestimated the relevance of State Machine and Deployment Diagrams. While the former is semantically rich, but often rejected by novices, the latter is used at

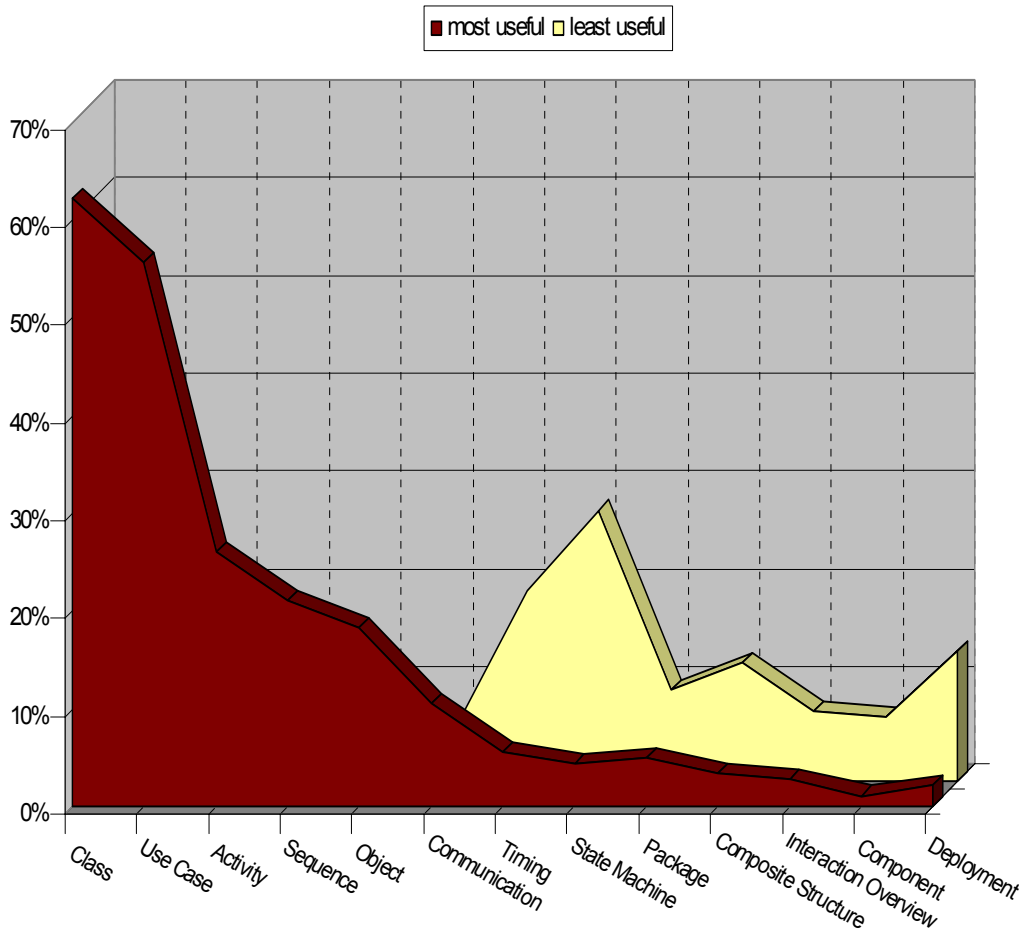
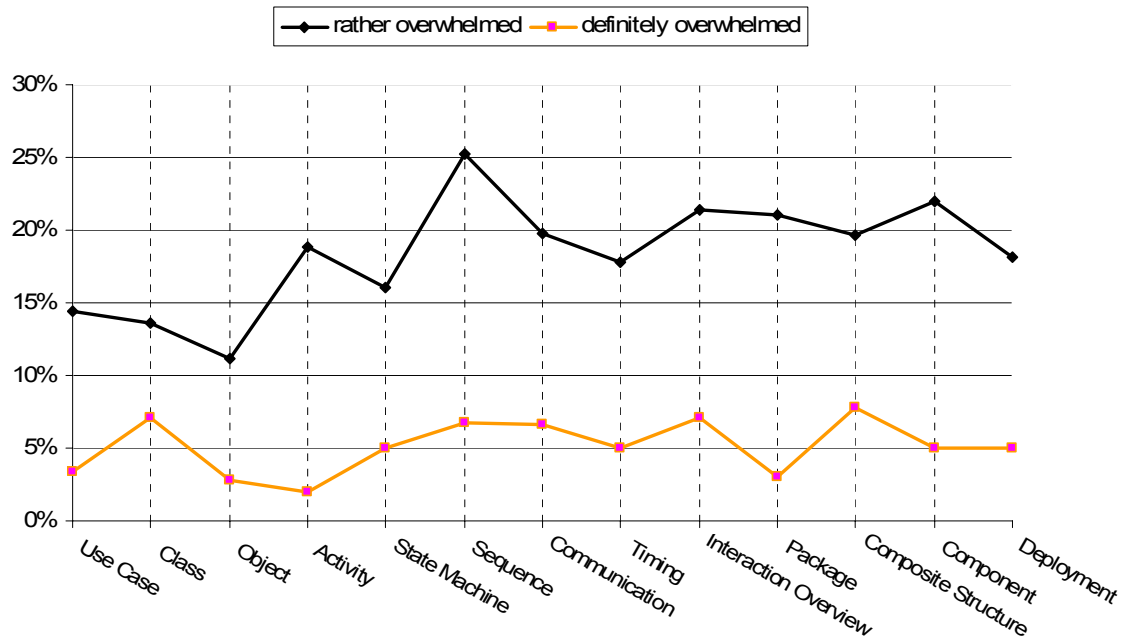


Figure 3: Usefulness of the specific UML diagrams

the lower, closer to implementation, disciplines of system development process. Therefore, the teaching of these types of diagrams could be transferred to the object-oriented programming courses.

### **Diagrams Types and Their Modeling Constructs**

As concerned the fourth criterion, students were supposed to enumerate diagrams particularly overwhelmed with UML notions (Figure 4). Most interaction diagrams were found on the list. Sequence Diagram was considered overwhelmed or very overwhelmed with specific modeling constructs by 32% of the interviewees, while Interaction Overview Diagram and Communication Diagram by 28% and 27% respectively. Only Timing Diagram was ranked as average. On the other hand, number of UML notions used while creating a diagram was not a problem in the case of Object Diagrams, Use Case Diagrams and Class Diagrams. Only respectively - 14%, 18% and 20% of the respondents respectively mentioned these diagrams as overwhelming. The case of Class Diagrams may be considered as an interesting one. This type of diagram, in fact is a complex one, with relatively large number of modeling constructs. However they are accepted and naturally mastered by students because they are aware of the significance of the classes in contemporary programming languages.



**Figure 4: UML diagrams overwhelmed with modeling constructs**

### **User-Friendliness of UML Diagrams**

User-friendliness is one of the keywords and challenges of the computing field. Assessment of UML diagrams under this angle should facilitate the specification of UML Light version. Definitely the Use Case Diagram was recognized as the most easy to use in the family of 13 UML diagrams (Figure 5). The survey respondents (74%) confirm this feature, so required at the high level of system specification. This aspect of the system model should be as precise as possible, remaining easy to interpret by all system stakeholders, in particular system owners, managers and future users. Acknowledged user-friendliness of Use Case Diagrams is a good starting point for

achieving system specification correctness, precision, consistency and completeness by using the other related UML diagrams, supporting Use Case Diagrams.

Due to the pragmatic role of Class Diagrams for programming, they have also achieved a high rank of acceptance - 66% of the respondents classified this diagram as an easy or very easy. Students appreciated (59%) the significance of Activity Diagrams as a backbone of algorithms and programs. Certain types of UML diagrams should be reconsidered in respect of their “user-friendliness”. In particular, Interaction Overview Diagrams were classified as difficult or very difficult to use by 43% of the students. Also Deployment Diagrams (39%) and Composite Structure Diagrams (38%) were found difficult to use. Therefore, the mentioned diagrams are the candidates for excluding them from the UML 2.x Light version.

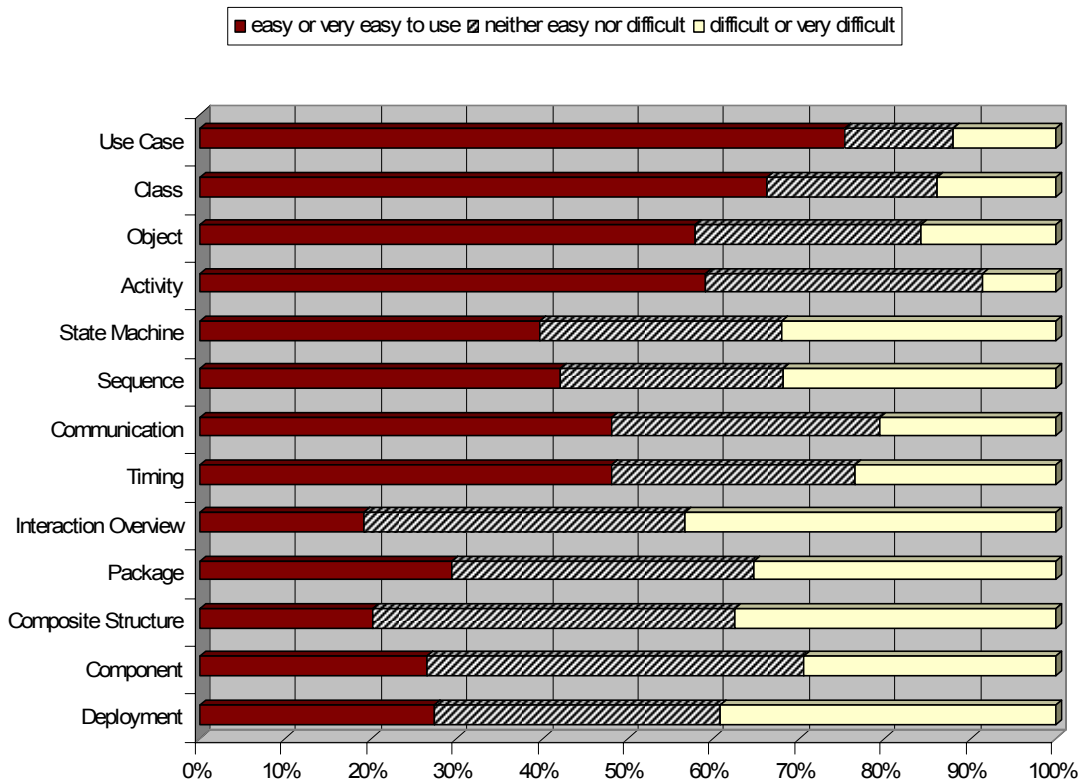


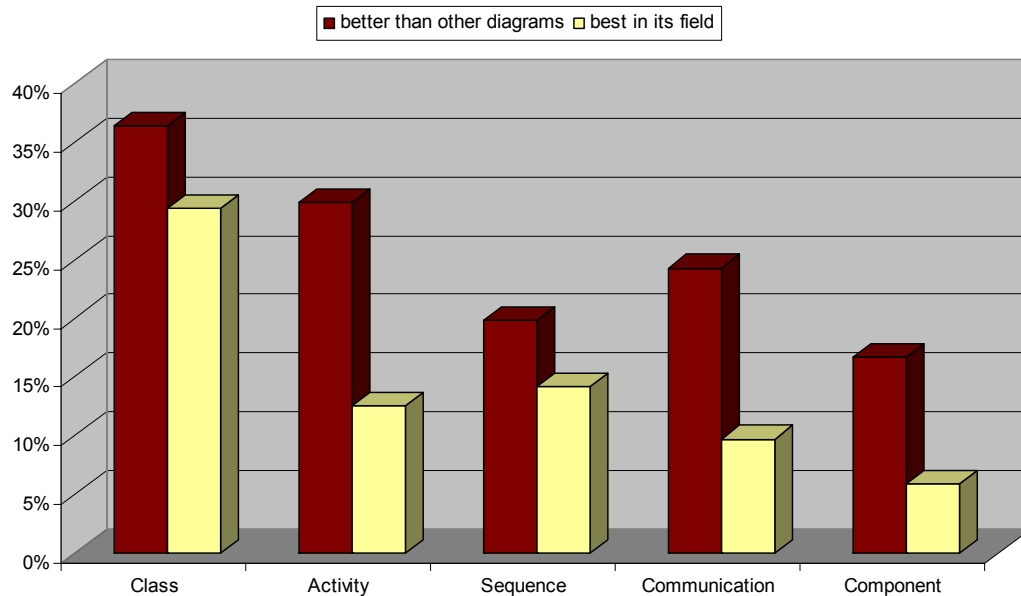
Figure 5: Assessment of user-friendliness of UML diagrams

### UML Diagrams for Code Generating

The development of CASE tools inspired the research and works on source code generation on the basics of system documentation. UML diagrams at large, give the profound opportunity for code generation on the basis of precise system specifications. The interviewees assessed the following types of diagrams as a good basis for code generation:

- Class Diagrams (66% total);
- Activity Diagrams (42%);
- Sequence Diagrams (34%);
- Communication Diagrams (34%);
- Component Diagrams (23%).

Again the Class Diagrams have been recognized as the most helpful types of UML diagrams while transferring system model into a code (Figure 6). The contribution and usefulness of the other UML diagrams in respect of code generation, but not included in the above group of five types, have been estimated as low.



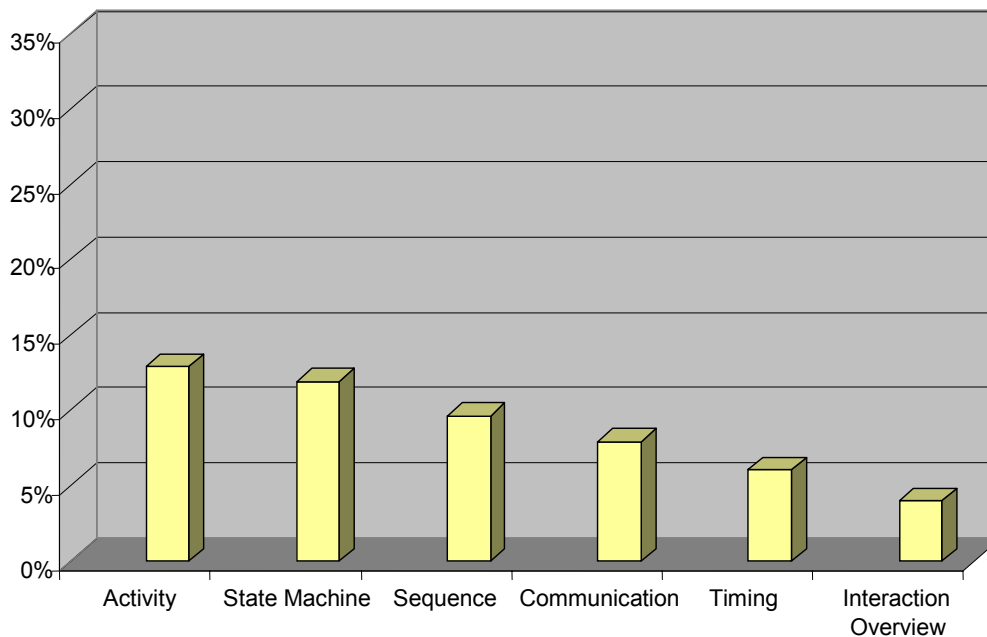
**Figure 6: UML diagrams helpful for the source code generating**

### ***Modeling the System Dynamics***

Potential UML user has quite a number of UML diagrams types at his/her disposal, used for describing system dynamics. Some of them are relatively intuitive and easy to use (eg. Activity Diagrams, Timing Diagrams) while the others are very precise, robust and consequently difficult, but they still remain helpful and are eagerly used by system analysts and designers. In particular, Sequence and Communication Diagrams. Diagrams are not so intuitive as the types of diagrams used for modeling system requirements, as they are addressed to professionally experienced programmers. Precision in developing system low-level dynamics specifications as well as their transferability to the source code should be the deciding factors of their functionality. As shown at Figure 7, besides Interaction Overview Diagrams, all remaining UML dynamics diagrams are helpful in preparing such specifications. Activity Diagrams were considered as the best in this area by as much as 13% of the respondents. Given the fact that Activity Diagrams are rather user-friendly, the closest to the structured methodologies, they remain a good basis for specifying the system logic and source code backbone.

### **Summary**

The survey results presented this paper are helpful in defining the scope of UML 2.x Light version. Such version would be extremely stimulating and motivating in effective teaching of UML 2.x. This concept was warmly welcomed by students and still does not limit the UML potential. The system specifications elaborated in the Light version could be subsequently extended towards the complete systems by the use of full scope of UML modeling notions and diagrams.



**Figure 7: UML diagrams for supporting system dynamics specification**

To sum up, the following UML diagrams were selected and indicated in the survey as the components of the UML Light version:

- Use Case Diagrams,
- Class Diagrams,
- Activity Diagrams,
- Sequence Diagrams.

These four types of diagrams (Figure 8) enable modeling of all essential system aspects, i.e. system requirements, analysis and design of system structure and dynamics. This conclusion was revealed by the first criterion analyzed in the reported investigation and than consequently supported by six succeeding criteria.

Not all modeling constructs are used while preparing the system specifications according to the UML 2.x Light version. Students are particularly overwhelmed by the number of modeling notions mostly while developing Sequence Diagrams and Activity Diagrams. Therefore, only the most relevant of these diagrams notions should be transferred to the UML 2.x Light version. Wrycza, Marcinkowski, and Wyrzykowski (2005) divided the UML modeling notions into basic and advanced ones. The proposal of the division of the specific modeling constructs adequate for the four selected types of diagrams respectively is presented in Table 1.

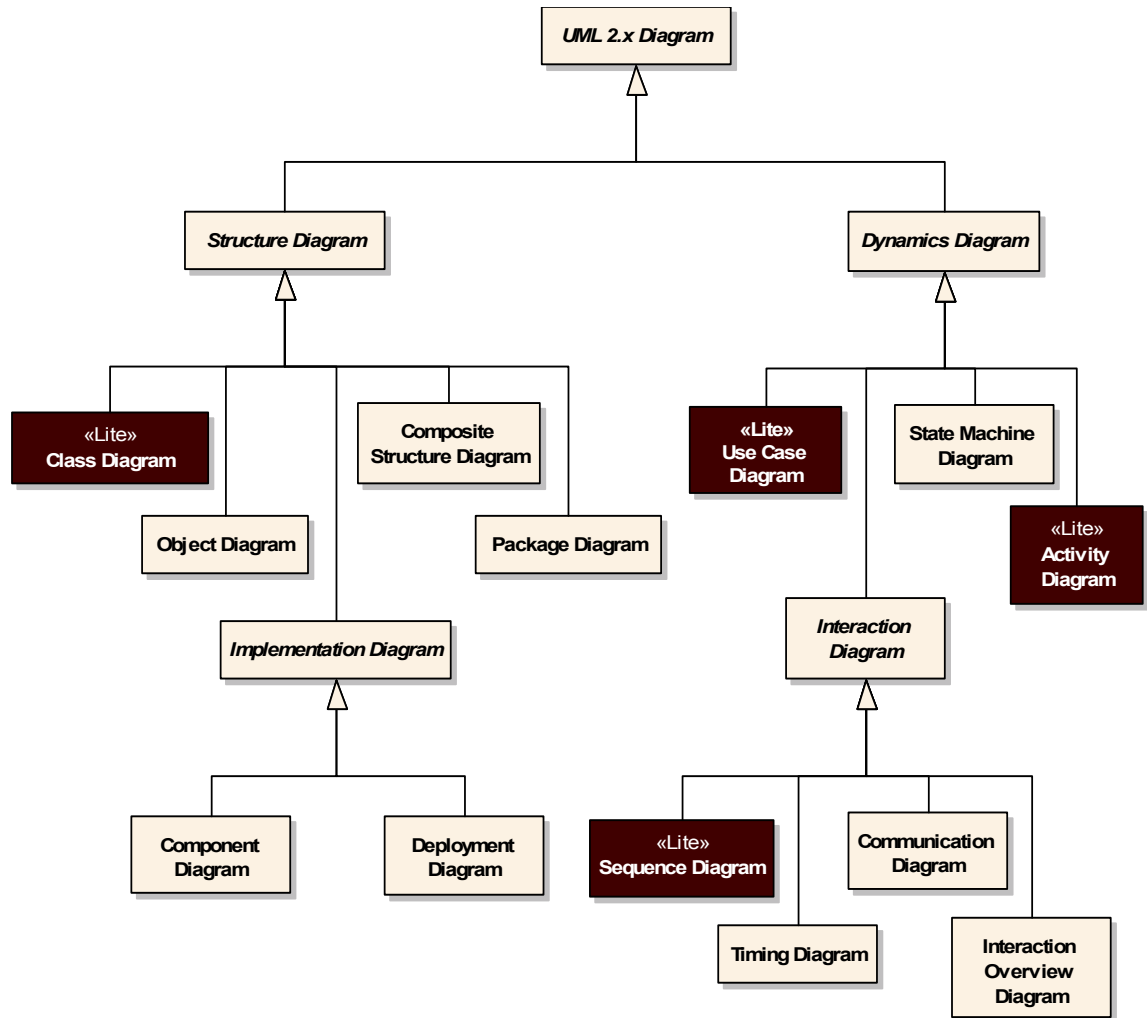


Figure 8: UML 2.x selected for the Light version

Table 1: The basic and advanced modeling constructs in respect of UML 2.x Light version.

	Class Diagram	Use Case Diagram	Activity Diagram	Sequence Diagram
Basic notions	Class Attribute Operation Binary association Association name Role name Navigability Multiplicity Aggregation Composition	Use case Actor Binary association	Activity Subactivity Activity Initial Activity Final Control Flow	Actor Class Boundary class Control class Entity class Lifeline Execution specification Synchronous message

Advanced notions	Responsibility Visibility Static attributes/operations N-ary associations Association classes Reflexive associations Multiple associations Qualification Generalization Dependency Realization	«include» dependency «extend» dependency Generalization Types of actors Multiplicity Navigability Realization	Decision Activity edge connector Merge node Action Pin Activity parameter node Weight Signal Central buffer Data store Activity partition Expansion region Interruptible activity region Exception handler	Asynchronous message Return message Lost message Found message Balking message Timeout message Guard condition Message to self Iteration Branching Interaction fragment Interaction occurrence Gate
------------------	--	---	---	---

Both four selected types of UML diagrams (Class, Use Case, Activity and Sequence Diagrams), shown at Figure 8 as well as respective basic modeling categories of these types of diagrams (Table 1) form the proposed scope of UML 2.x Light version according to the survey accomplished.

## References

- Ambler, S. W. (2005). *The elements of UML 2.0 style*. Cambridge: Cambridge University Press.
- Booch, G., Rumbaugh, J. & Jacobson, I. (2004). *The UML reference manual* (2nd ed.). Boston: Addison-Wesley.
- Burton, P. J. & Bruhn, R. E. (2004). Using UML to facilitate the teaching of object-oriented systems analysis & design. *Journal of Computing Sciences in Colleges*, 19.
- DeLooze, L. L. (2005). Minimal UML diagrams for a data-driven web site. *SIGITE*.
- Dobing, B. & Parsons, J. (2006). How UML is used. *Communications of ACM*, 49.
- Flint, S., Gardner, H. & Boughton, C. (2004). Executable/translatable UML in computing education. *Conferences in Research and Practice in Information Technology*, 30. Raymond Lister and Alison Young Ed.
- Jacobson, I., Christerson, M., Jonsson, P. & Overgaard, G. (1992). *Object-oriented software engineering: A use-case driven approach*. Boston: Addison-Wesley.
- Kontio, M. (2005). Architectural manifesto: Designing software architectures. Part 5: Introducing the 4+1 view model. Retrieved from <http://www.omg.org/cgi-bin/doc?formal/05-07-04>
- OMG (2005). Object Management Group. Unified modeling language 2.0 superstructure specification. Retrieved from <http://www.omg.org/cgi-bin/doc?formal/05-07-04>
- OMG (2006). Object Management Group. The UML 2.1 superstructure convenience document. Retrieved from <http://www.omg.org/cgi-bin/doc?ptc/2006-04-02>
- Trujillo, J. (2006). A report on the first international workshop on best practices of UML. *SIGMOD Record*, 35.

- Wrycza, S. & Marcinkowski, B. (2005). UML 2 teaching at postgraduate studies – Prerequisites and practice. *Proceedings of ISECON 2005, 22, New Orleans*. AITP Foundation for Information Technology Education.
- Wrycza, S. & Marcinkowski, B. (2006). UML 2 academic course – Methodological background and survey benchmarking. *Proceedings of ISECON 2006, 23, Dallas*. AITP Foundation for Information Technology Education.
- Wrycza, S., Marcinkowski, B. & Wyrzykowski, K. (2005). Systems modeling with UML 2 (in Polish). *Helion, Gliwice, pp. 1-456*.

## Biographies



**Stanislaw Wrycza** is a Professor and Head of Department of Information Systems at University of Gdansk, Poland. His main areas of research and teaching activities are: information system development, UML, databases, e-business, e-learning. He has written or edited 25 books, published over 150 articles in English and Polish in professional journals and papers at international conferences. He was a first President of Polish Society for Information Systems Research (1995-2000) and chairman of Program and Organizational Committees of the Xth European Conference on Information Systems (ECIS2002), held in Gdansk in June 2002. He is the founding member of the Association for Information Systems (AIS) and Information Systems Aca-

ademic Heads International (ISAHI) as well as the President of Polish Chapter of AIS (PLAIS) since 2006.

Prof. Stanislaw Wrycza was co-initiator of the international conference on Information Systems Development (ISD). In its 14th editions several hundred papers from over 50 countries were published, recently by Springer Science. Prof. Wrycza has participated in numerous (over 100) program committees of international conferences and has been the member of editorial board of several international IS journals like Information Systems Journal or Information Systems and eBusiness. He also has taken part in European educational and research and educational programs of the European Union. For more information see: <http://www.wrycza.wzr.pl/indexen.html>

**Bartosz Marcinkowski**, MSc, is assistant at the Department of Business Informatics at the University of Gdansk, Poland. He is author and co-author of numerous publications in Polish and English, a member of Pomeranian Regional Cisco systems Networking Academy, and an instructor involved in European projects.