Towards a Light Version of UML 2.X: Appraisal and Model

Stanisław Wrycza, Bartosz Marcinkowski

Piaskowa 9, 81-824 Sopot, Poland; swrycza@univ.gda.pl; bmarc@univ.gda.pl

UML 2.x version has become an even more complicated and diverse set of graphical techniques than its predecessors. Therefore, system developers propose preparation of its reduced, limited or minimal version called Light UML. This problem has become also a serious challenge for the UML academic teachers. The goal of this paper is the study of specifying the UML 2.x Light version content on the basis of the questionnaire survey registering opinions of 180 university students of the University of 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 UML 2.x version, resulting from the accomplished survey, is exposed in the last part of the paper.

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

Za uporabo Light verzije UML 2.x-a: ocena in vzorec

UML 2.x verzija je postala celo bolj zapleten in raznolik skupek grafičnih tehnik kot njeni predhodniki. Zato sistemski razvijalci predlagajo razvoj zmanjšane, omejene ali minimalne verzije imenovane Light UML. Ta problem predstavlja tudi resen izziv za akademske kroge, ki se ukvarjajo s poučevanjem UML. Namen tega prispevka je preučitev in opredelitev vsebine UML 2.x Light verzije na osnovi ankete, s katero smo zbrali mnenja 180 študentov na Univerzi v Gdansku. Uvodu sledi razlaga metodoloških zahtev raziskave. Zatem so predstavljeni rezultati, ki so interpretirani skladno s sedmimi bistvenimi ocenjevalnimi kriteriji UML diagramov, ki so bili vključeni v anketo. Končna UML 2.x verzija, ki izhaja iz izvedene raziskave, je razložena v zadnjem delu prispevka.

Ključne besede: UML 2.x, verzija UML Light, poučevanje UML, anketna raziskava, uporabniški diagrami, razredni diagrami, sekvenčni diagrami, diagrami aktivnosti

1 Introduction

Unified Modeling Language (UML), proposed by G. Booch, I. Jacobson and J. 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 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 stress that the use of strict subsets of UML is easier to understand than the full language notation. Burton & Bruhn (2004) generalize their experiences related to use of the UML and un-

derline 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). Another survey, carried out among 171 practitioners, was directed at the UML version that would have a limited scope as well (Dobing & Parsons, 2006). It seems that the quickness of UML upgrading and implementing 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 & Marcinkowski, 2005). The authors identified and analyzed several problems described in (Wrycza & 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), complicated interrelationships among them and the extensive number of modeling notions. 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 ver-

Table 1. Excerpt from course curriculum

Topic	**Number of hours**	
	Lectures	**Labs**
UML – development, structure, terms	1h	
Use Case Diagram	3h	4h (incl. UC scenarios)
Class Diagram and Object Diagram	4h	2h
Activity Diagram	3h	2h
State Machine Diagram	2h	1h
Introduction to Interaction Diagrams	1h	
Sequence Diagram	3h	3h
Communication Diagram	1h	1h
Timing Diagram	1h	1h (optionally)
Interaction Overview Diagrams	1h	1h (optionally)
Implementation Diagrams	2h	
Composite Structure Diagram	1h	
Package Diagram	1h	
Rational Unified Process	2h	
Business Modeling with UML profile	2h	
Robustness Analysis	1h	
Computer-Aided Software Engineering	1h	1h

Source: Wrycza & Marcinkowski (2005)

sion by the application of complete scope of UML modeling diagrams and constructs.

2 Methodological background

To solve the problem of UML Light version concept, the authors decided to carry out the questionnaire survey among the university students. The target group encompassed 180 students from public as well as private universities, 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 "UML 2.0 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 course curriculum in its current shape includes both lectures and laboratories, as presented in Table 1.

As noted above, the appropriate questionnaire containing 17 basic questions was elaborated and handed to 180 students taking part in UML course. The questions were focused around Light version concepts, reciprocal influence of structured and object-oriented approach as well as possible UML extensions. The questions arose on the basis of didactic experience of teachers, problems observed during the laboratories as well as questions and suggestions reported by the students. To make the proper assessment of the UML 2.x Light version the following seven crucial issues, raised in questionnaire, were analyzed:

1. UML complexity level,
2. UML diagrams cardinality,
3. usefulness of the specific diagrams,
4. choice of diagrams overwhelmed with modeling constructs,
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.

3 Selected results of the survey

UML 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

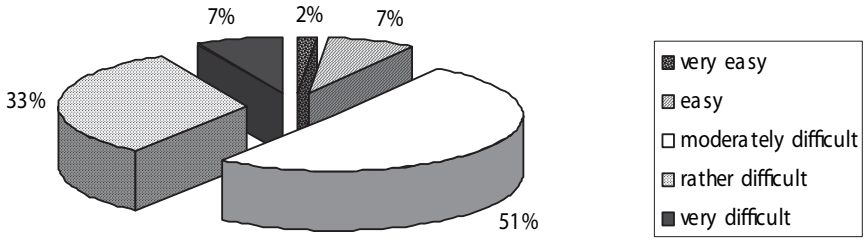

Figure 1. UML complexity level

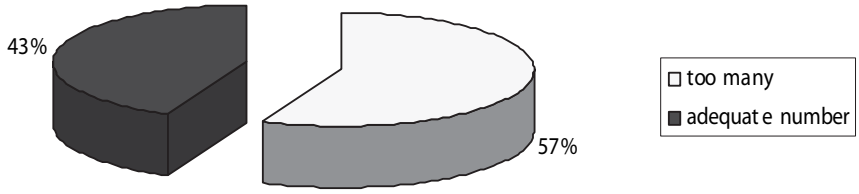

Figure 2. Adequacy of the number of UML diagrams

respondents would welcome the more introductory, i.e. the Light version of UML.

UML diagrams cardinality

The students taking part in the survey had a chance to exercise 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 at Figure 2. The remaining respondents accepted all types of diagrams, not assessing however the potential surplus of cardinality of modeling notions that were used in each type of diagram.

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 diagram types 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 of information system respectively. Supplementary, Use Case Diagrams initiate iterative- incremental 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 the opinion of teachers, students underestimated the relevance of State Machine Diagram and Deployment Diagram. While the former is semantically rich, but often rejected by novices, the latter is used at 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 mode-

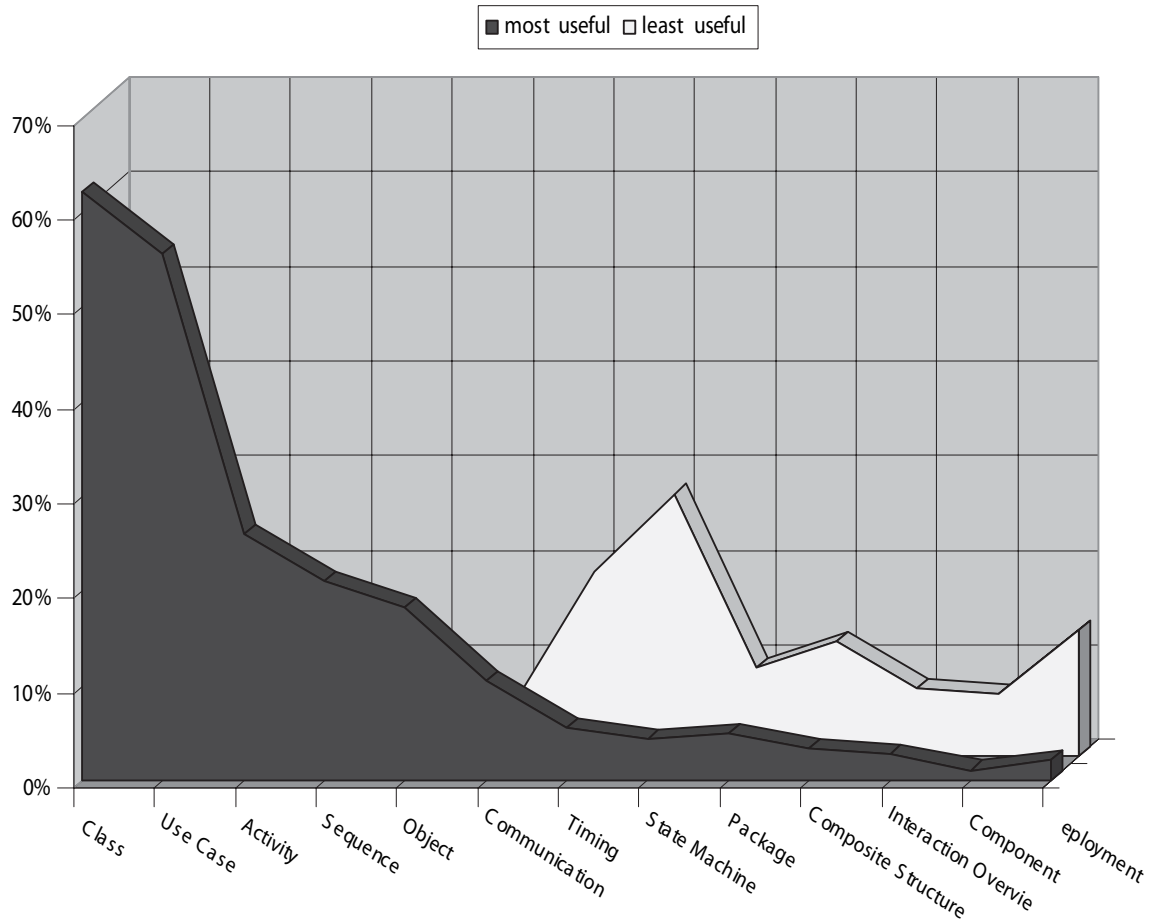

Figure 3. Usefulness of the specific UML diagrams

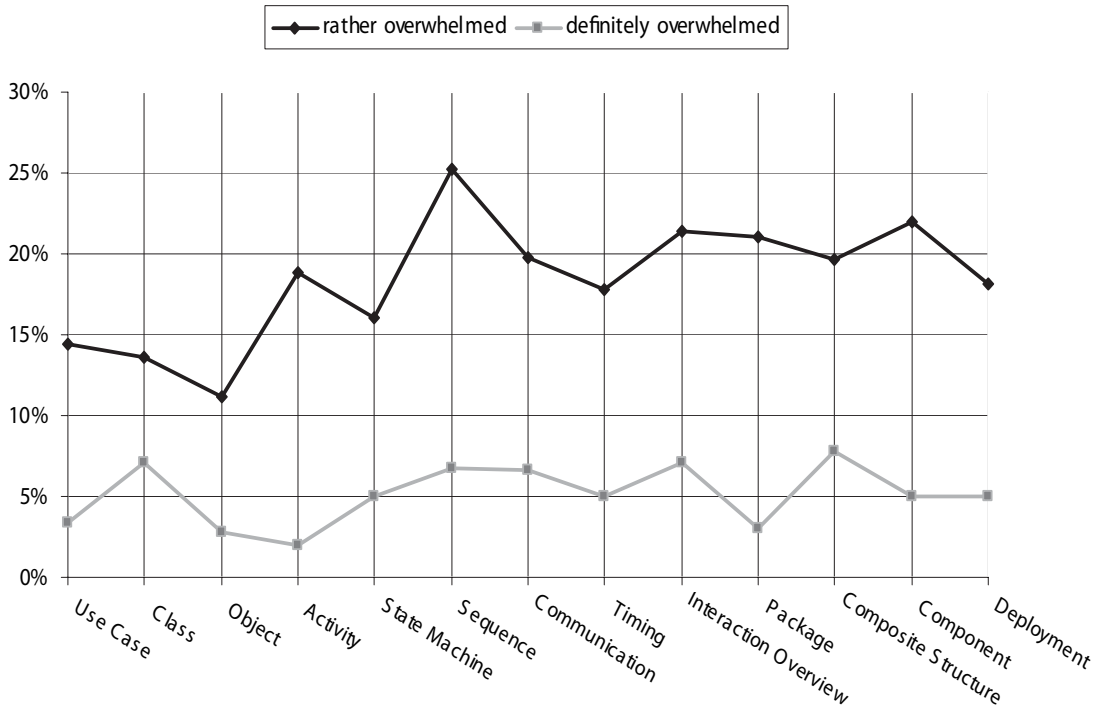

Figure 4. UML diagrams overwhelmed with modeling constructs

ling 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 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 is in fact a complex one, consisting of a relatively large number of modeling constructs. However they are accepted and naturally mastered by students, owing to the awareness of the significance of the classes in contemporary programming languages.

User-friendliness of UML diagrams

User-friendliness is one of the keywords and challenges of 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 type of diagram as an easy or very easy one. Students appreciated (59%) the significance of Activity Diagrams as a backbone of algorithms and programs. Certain types of UML diagrams ought to 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 natural candidates for excluding them from the scope of the UML 2.x Light version.

UML diagrams best-suited for source code generation

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 particularly good basis for code generation:

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

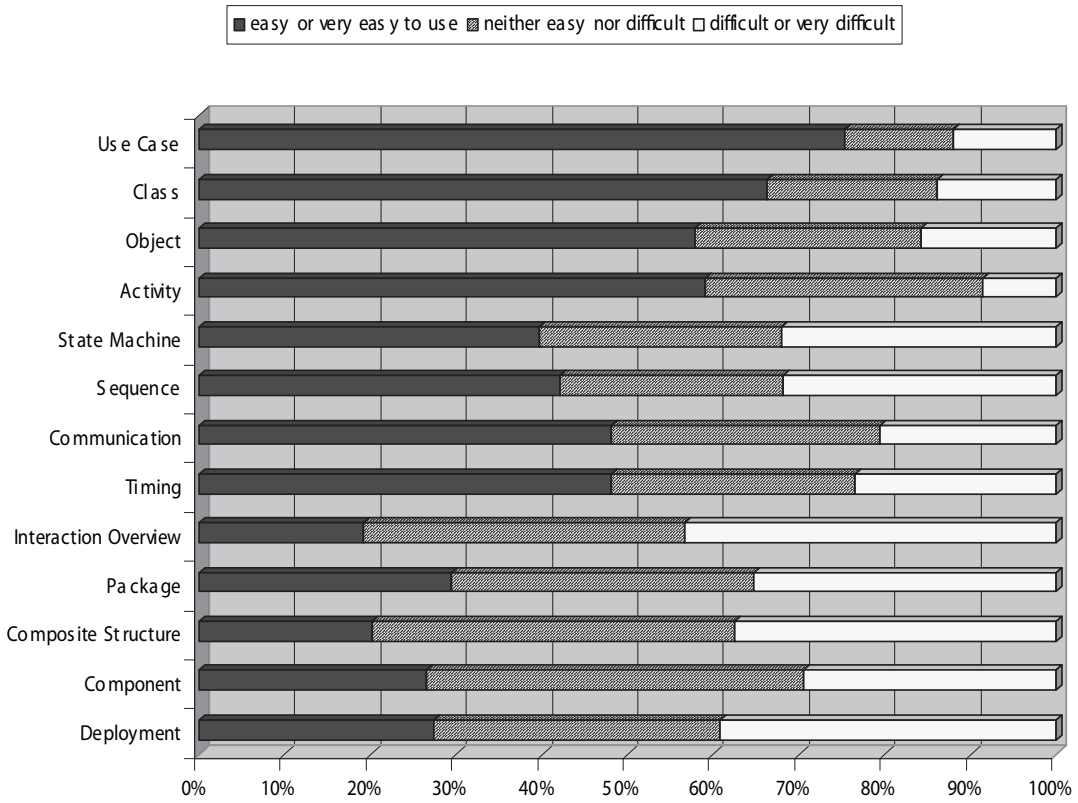

Figure 5. Assessment of UML diagrams user-friendliness

Again the Class Diagrams have been recognized as the most helpful types of UML diagrams while transferring system model into a code (Figure 6). Both 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.

Modeling the system dynamics

Potential UML user has quite a number of UML diagrams types used for describing system dynamics at his/her disposal. Some of them are relatively intuitive and easy to use (eg. Activity Diagrams, Timing Diagrams) while the others are very precise, robust and consequently

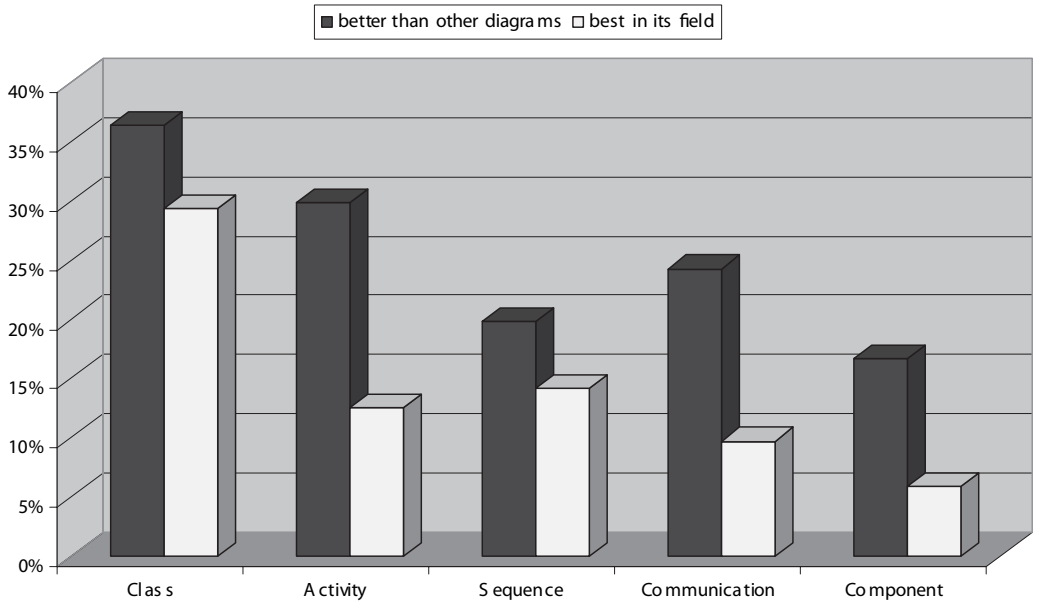

Figure 6. UML diagrams best-suited for source code generation

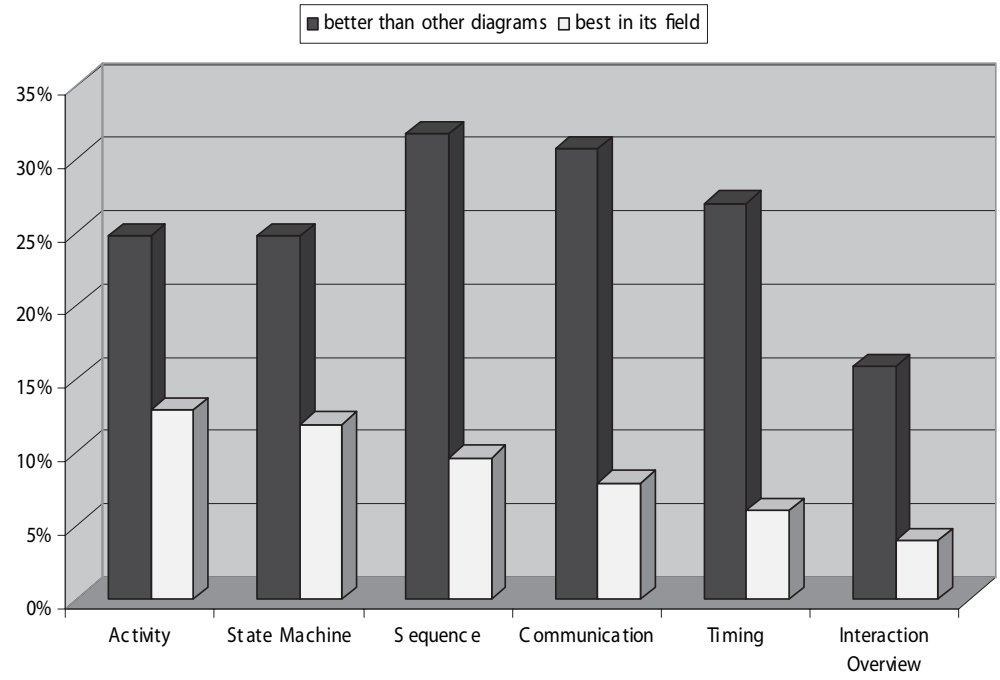

Figure 7. UML diagrams for supporting system dynamics specification

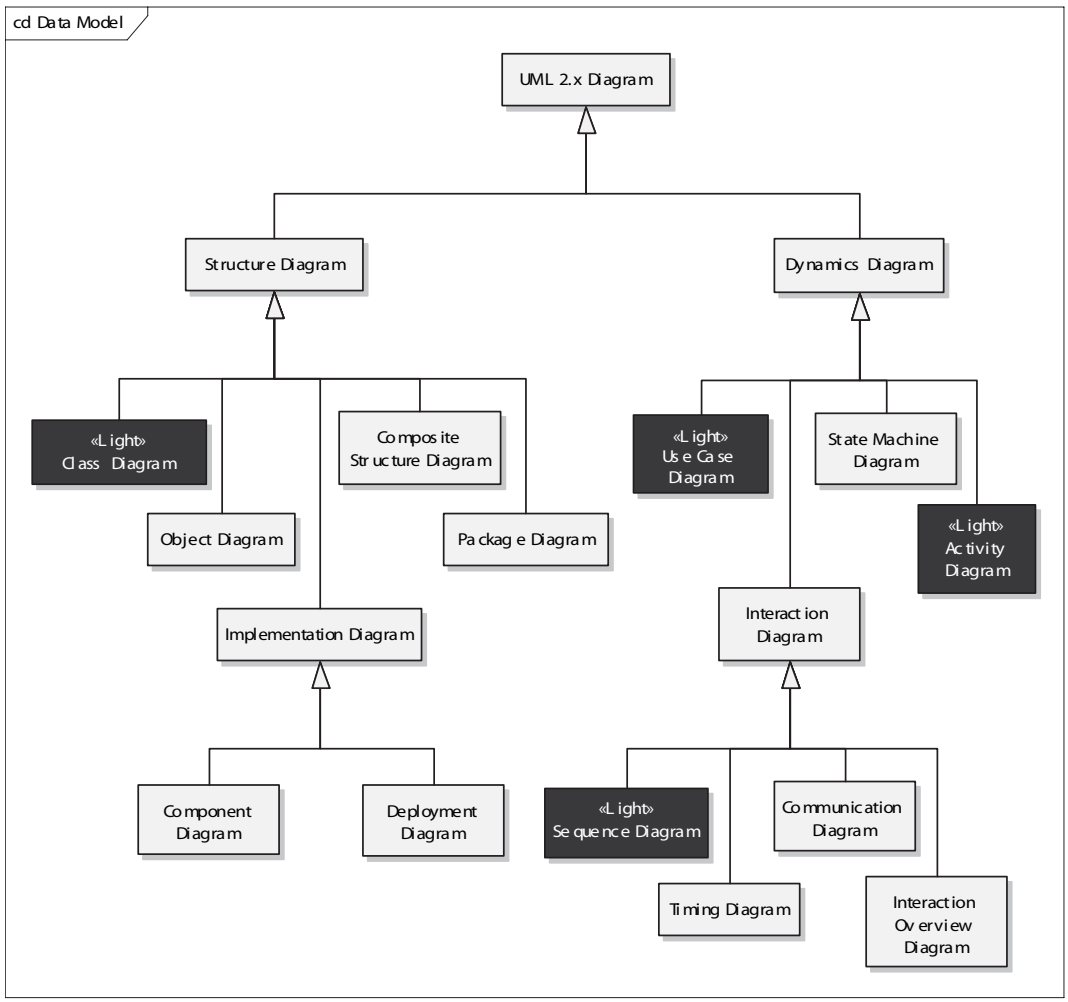

Figure 8. UML 2.x diagrams selected for the Light version

Table 2. 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

difficult, but they still remain helpful and are eagerly used by system analysts and designers. In particular, Sequence and Communication Diagrams are not as intuitive as diagrams used for modeling system requirements, by and large because they are addressed to professional and experienced programmers. Precision in developing low-level system 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 the best in this field 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.

4 Summary

The survey results presented in this paper are helpful in defining the scope of the 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 using the Light version could be subsequently extended towards the complete systems by the implementation of full scope of UML modeling notions and diagrams.

To sum up, the following UML diagrams were selected and indicated in the survey as the components of the proposed 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 survey and then 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 & 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 2.

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. The survey results have had an influence on the final curriculum of the UML course taught by the authors and presented in Table 1. The outline of the lectures was decided to remain unchanged. The laboratories will include exclusively the types of diagrams identified in the questionnaire survey and selected for the Light version. Thus, the distribution of lab hours will change accordingly to the conclusions of the survey:

- Use Case diagrams – 4hrs,
- Class and object diagrams – 4 hrs,
- Activity diagrams – 3 hrs,
- Sequence diagrams – 4 hrs.
- Such modifications should meet the expectations of students that were revealed by the survey.

References

Ambler, S.W. (2005). *The Elements of UML 2.0 Style*. Cambridge University Press, Cambridge.

Booch, G., Rumbaugh, J. & Jacobson, I. (2004). *The UML Reference Manual. 2nd Edition*. Addison-Wesley, Boston.

Burton, P.J. and 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*. Addison-Wesley, Boston.

Kontio, M., Architectural Manifesto: Designing Software Architectures. Part 5. Introducing the 4+1 View Model, Available from: http://www-128.ibm.com/developerworks/wireless/library/wi-arch11 (18. September 2005).

Kruchten, P. (1995). Architectural Blueprints – the "4+1" View Model of Software Architecture. *IEEE Software*, **12**.

Object Management Group, The UML 2.1 Superstructure Convenience Document, Available from: http://www.omg.org/cgi-bin/doc?ptc/2006-04-02 (2. April 2006).

Object Management Group, Unified Modeling Language 2.0 Superstructure Specification, Available from: http://www.omg.org/cgi-bin/doc?formal/05-07-04 (4. July 2005).

Trujillo, J. (2006). A Report on the First International Workshop on Best Practices of UML. *SIGMOD Record*, **35**.

Wrycza, S., Marcinkowski, B. & Wyrzykowski, K. (2005). *Systems Modeling with UML 2* (in Polish). Helion, Gliwice.

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.

Stanisław Wrycza is professor and Head of Department of Business Informatics at University of Gdansk, Poland. He is the author of over 150 books, papers and articles published in English and Polish. He has organized many international conferences, workshops and symposia including Xth European Conference on Information Systems ECIS2002 in Gdansk. He is involved in the research, teaching and publishing in information systems development, UML, databases, e-business and e-learning.

Bartosz Marcinkowski is Assistant at the Department of Business Informatics at University of Gdansk, Poland. His research and teaching interests encompass information systems development as well as computer networking.