Student Experience with Software Modeling Tools

Luciane T. W. Agner[1] , Timothy C. Lethbridge[2], Inali W. Soares[1]
lagner@unicentro.br, tcl@eecs.uottawa.ca, inali@unicentro.br

[1]Department of Computer Science,
Mid-West State University (UNICENTRO),
Guarapuava, Brazil

[2]Electrical Engineering and Computer Science,
University of Ottawa,
Ottawa, Canada

Abstract

Modeling is a key concept in software engineering education, since students need to learn it in order to be able to produce large-scale and reliable software. Quality tools are needed to support modeling in education, but existing tools vary considerably both in their features and in their strengths and weaknesses. The objective of the research presented in this paper was to help professors and students choose tools by determining which strengths and weaknesses matter most to students, which tools exhibit which of these strengths and weaknesses, and how difficult to use are various tools for students. To achieve this objective, we conducted a survey of the use of modeling tools among students in software engineering courses from Brazil, Canada, the USA, Spain, Denmark, the UK and China. We report the results regarding the 31 UML tools that 117 participants have used, focusing on the nine tools that the students have used most heavily. Common benefits quoted by students in choosing a tool include simplicity of installing and learning, being free, supporting the most important notations and providing code generation. The most cited complaints about tools include lack of feedback, being slow to use, difficulty drawing the diagrams, not interacting well with other tools, and being complex to use. This research also compares the results with the findings of another survey conducted among professors who taught modeling. The results should benefit tool developers by suggesting ways they could improve their tools. The results should also help inform the selection of tools by educators and students.

1. Introduction

Software modelling is a core topic in teaching Software Engineering, since modeling is increasingly important when developing high-quality software [1]. The use of modeling tools is important to ensure students can learn to model comprehensively and receive effective feedback about their models, including by being able to turn their models into actual executable systems [2]. Students who are just taught to model as if it were documentation, either without tools or with simple drawing tools, are at a disadvantage entering the job market.

To teach effectively using modeling tools, professors need to know their strengths and weaknesses when selecting them for their students to use. Professors need to select tools that strike a good balance between power, availability and complexity, most likely with different tools being selected at different levels of education. A professor can look at the documentation of each tool, and can also download and try most of them; however, knowing the negative and positive opinions that others have of each tool should also help guide their selection.

If students are free to choose their own tool, then students themselves would also benefit from knowing the strengths and weaknesses identified by their peers.

Additionally, tool developers also need to know what attributes of tools are perceived to be most beneficial, or to cause the greatest frustration, so they can address weaknesses of their tool, and try to replicate the strengths of competing tools.

In this paper, we discuss results of a survey of students, highlighting strengths and weaknesses of software modeling tools as used in their education. Almost all tools discussed in this survey enable modeling in the Unified Modeling Language (UML). This was not a deliberate decision; it simply proved to be the case that the most-used tools are all ones that work with UML. UML is considered as de-facto standard for software modeling and is widely used in education and industry [3,4].

Our survey involved students who had taken software engineering courses that covered software modeling. The courses were offered in seven countries: Brazil, Canada, USA, Spain, Denmark, UK and China. We investigated which modeling tools have been used by the students, and their perceptions about these tools. The top nine most-heavily-used tools were analyzed more deeply, identifying their main benefits and drawbacks according to the students.

In a previous paper [5], we presented a similar survey covering perceptions of professors of software engineering (SE) courses from universities around the world. We summarise the results of that survey in Section 2.1. Some of the participating professors helped us to circulate the survey discussed in this paper to their students. In addition to presenting the results of the students' survey, we give comparisons between perceptions of professors and students.

To summarize: The goal of this survey was to determine the main benefits of modeling tools that students believe have helped them in their process of learning modeling, the main drawbacks that bother students regarding modeling tools, as well the extent to which students feel widely-used modeling tools have these benefits or drawbacks.

The paper is structured as follows. Section 2 presents the related work, including our survey of professors; Section 3 details the method used to perform the survey, including the specific questions we asked to achieve our goals; Section 4 presents the results of the survey; Section 5 discusses the results; Section 6 presents some of the results in the form of a ‘Magic Quadrant’, and Section 7 discusses potential threats to validity. Finally, Section 8 presents conclusions and gives suggestions for further work.

2. Related Work

In this section, we provide an overview of other research into the use of tools for software modeling in education.

2.1 A Survey of Professors’ Use of Modeling Tools

In our own prior research, conducted in late 2016 [5], we obtained detailed data from 125 professors in 30 countries and all populated continents, to determine which tools they used to teach modeling, and what they perceived to be the advantages and disadvantages of these tools.

All the professors had taught software modeling to undergraduates in the previous 5 years. Among these professors, the tool they used most was ArgoUML (36.6% had used it), with five additional tools being used by more than 20%: Visual Paradigm, StarUML, Papyrus, Astah and Magic Draw.

Four benefits provided the greatest motivation for professors to use particular tools: being simple to learn, being simple to install, being free and supporting the most important notations.

The most easy-to-use tools, in the opinion of the professors, were PlantUML, Visio and Umple; the most difficult to use were BoUML, IBM Rational Rhapsody, Papyrus and Acceleo.

The most common complaints about tools were not interacting well with other tools, not supporting sufficient modeling aspects and being complex to use. The most complex of seven most-used tools were reported to be Magic Draw and Papyrus. Visual Paradigm and Astah were considered the least complex of this set.

In some of the discussion later in this paper, we compare the results from this prior survey of professors with the survey of students presented in this paper. The study of students reported in this paper uses the same questions, simply adjusted to fit the context such as changing ‘Teach’ to ‘Learn’.

2.2 Other Research Describing Experience with Modeling and Modeling Tools

In 2014, Reggio and colleagues [6] presented a survey of the use of UML diagram types, to find out which were known and which were used. They point out in the paper that UML is often criticized for being overly complex. Indeed, several types of UML diagrams were known by fewer than half of the survey participants, and most had been produced by fewer than half of the industry participants, academic participants (mostly professors and PhD students) in general knew and had produced more of the diagram types. Although this survey used a similar method to our own, it did not ask about benefits and drawbacks of the diagram types, and nor did it ask about use of tools. Our work in this paper therefore is orthogonal to Reggio’s survey.

In 2016, Kuzniarz and Martins conducted a pilot study on methods, techniques, and tools used in teaching Model-Driven Software Development (MDSD) [7]. This work analyzed topics such as modeling in UML, requirements modeling and model transformation. The participants were seven professors involved in teaching MDSD from Sweden, Poland, Spain, Canada and Brazil. The results identified ten tools used to teach modeling: Visual Paradigm, Astah, Eclipse Modeling Tools, OCL Editor, Papyrus, ATL, Acceleo, ArgoUML, Enterprise Architect, and Rational Software[1]. Visual Paradigm was the most cited tool, which was used by three of the professors. Also, group projects were the most adopted approach to help the students understand the concepts related to UML modeling. Since this was a pilot, the sample considered contains only a few professors, making it difficult to generalize the results. The research we conducted for this paper focuses on most of the same tools as Kuzniarz and Martins.

Tekinerdogan’s work from 2013 discusses Model-Driven Software Development topics based on his three years of experience teaching a graduate course on MDSD at Bilkent University in Turkey [8]. In that course, the students were allowed to use any tool of the Eclipse Modeling Project. He presents a discussion about the lack of maturity of the modeling tools. According to

[1] The exact Rational product was not identified.

the author, this lack of maturity was manifested as bugs, lack of documentation, and lack of support for integration with other tools.

Paige et al. discuss practices for teaching modeling based on their personal experience [9]. The results show that complexity of tools, lack of feedback and lack of resources (books and tutorials) are important complaints that inhibit the use of tools in teaching modeling.

It is important to note that our work reported in the previous subsection, as well as the above studies present only the perception of those who *teach* modeling. Prior work does not show the students' perception regarding the tools used.

One of us (Lethbridge) presented a survey of students who used the modeling tool Umple in their education [10]. The results show that the use of this tool helped students to understand UML concepts, and that the use of the Umple technology in the classroom made lecturing easier for professors. Other papers have also evaluated specific tools used in modeling education [11,12]. However, these works reflect unique experiences of the authors with certain tools.

Several papers deal with what aspects of the modeling process are supported by various tools [13,14,15,16]. In general, the conclusion from these papers is that tools do not sufficiently support all needed aspects of the modeling process due to their complexity or lack of maturity. The papers point out issues such as weak tool integration, lack of fit of tools to user's needs, poor user experience, and bad performance [13,14,15]. The literature clearly shows that there is a need for improved, more integrated, and comprehensive tool support for modeling [16]. However, this group of studies focus on modeling tool use in general and does not specifically address modeling education, nor do most of the papers identify which tools are most used.

3. Survey Method

The ethics application for the conduct of this survey was approved by The University of Ottawa Research Ethics Board. A pilot study was also conducted to obtain feedback and help us identify ambiguities, and improvements to wording. The final questionnaire was adapted to resolve these issues.

The main research questions we sought to answer, in order to address the goal outlined in the introduction, were:

- RQ1: *What are the main benefits found in modeling tools that help students learn about modeling?*

- RQ2: *What are the main drawbacks of modeling tools that bother students?*
- RQ3: *To what extent do widely-used modeling tools exhibit each of the benefits and drawbacks, in the opinion of students?*
- RQ4: *How difficult to use, in general, do students find widely-used modeling tools?*

3.1 Sampling

A total of 136 students responded by accessing the online survey, however only 117 gave detailed responses about the tools they use (the remainder gave us partial answers, consisting mostly of just demographic data).

Participants represented 7 countries: Brazil (36), Canada (35), USA (15), Spain (20), Denmark (7), UK (2) and China (2), distributed as shown in Figure 1. One participant did not give their country. Thus, the sample includes representatives from South America, North America, Europe and Asia. It covers fewer countries than the survey of professors discussed in Section 2.1, but still covers geographically dispersed regions,

The survey was directed at students who have learned modeling at some point in the last year or so, in order to focus on modeling tools that are currently being used, as opposed to outdated tools. The sample also included both undergraduate (56.1%) and graduate (43.9%) students who have taken software engineering courses that include modeling (Figure 2).

As we will discuss in Section 3, we asked the students detailed questions about the modeling tool they have used most and second most in their studies. Some students only answered regarding their most-used tool. Table I presents the number of thus-analyzed tools in each participating country. We believe that our sample is sufficiently representative because it has diversity in terms of geography, tools and student levels. Also, the results are similar to those obtained in our survey conducted with professors from 30 countries around the world [5] as mentioned in Section 2.1.

Although multiple tools and countries are covered, the concentration on specific universities means that it is not possible to infer from this survey the frequency of use of tools in the entire world student population. For that, it is necessary to refer to data from our survey of professors [5]. The results in this paper should, however, provide valid data regarding what students like and dislike about tools.

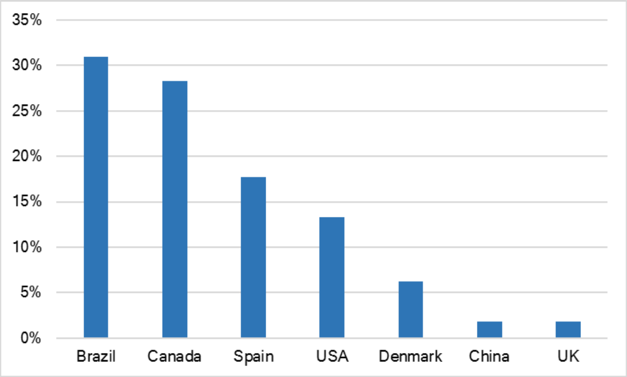


Figure 1. Proportion of students from each of the countries, from Question 2 of the survey

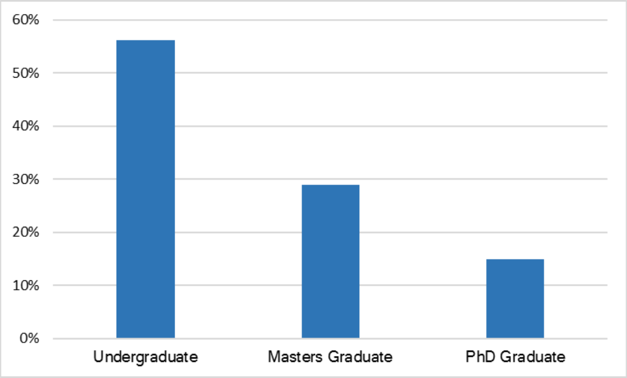


Figure 2. Level of study of the students, from Question 3 of the survey

TABLE I: **Tools for which we gathered detailed data from students, broken down by country**

Country	Number of students	Tools about which detailed responses were received, with the number of students giving such responses shown in brackets. Since 33% of students gave details responses about two separate tools, the numbers in parentheses sum to more than the number of students.
Brazil	36	ArgoUML (10), Astah (12), Dia (1), LucidChart(1), Papyrus (1), PlantUML (1), StarUML (13), Visual Paradigm (3)
Canada	35	ArgoUML (1), draw.io (1), Eclipse Modeling Tools (3), IBM Rational Software Modeler (2), jUCMNav (3), OmniGraff (1), OSATE (1), PlantUML (1), StarUML (2), Umple (20), Violet (1), Visio (3), Visual Paradigm (4), yED (1)
USA	15	Acceleo (1), ArgoUML (1), Cacoo (1), draw.io (4), OSATE (1), Simulink (1), StarUML (7), Visio (1), Visual Paradigm (1)
Spain	20	Magic Draw UML (19), Papyrus (5), USE (12)
Denmark	7	Eclipse Modeling Tools (6), yUML (1)
UK	2	Papyrus (2)
China	2	IBM Rational Software Modeler (1), Visio (1)
Total	117	(149 detailed responses about specific tools from the 117 students who identified their country)

The survey was conducted online from December 2016 to March 2017. The survey included an invitation explaining its objectives, instructions on how to fill in the questionnaire, as well as clarifications regarding risks and confidentiality. The invitation also informed participants that the estimated time for answering the questionnaire would be approximately 10 minutes. The students received the survey invitation through a professor who circulated the research request among their students. The professors had been participants in our survey of professors [5], discussed in Section 2.1.

3.2 Survey instrument

The survey questionnaire consisted of the 21 questions shown in Table II. The questionnaire covered student background (Q2-3), extent of use of modeling tools (Q4), tools they have *ever* used for modeling (Q5-6 – there may have been many), and tools they have *most heavily* used. They selected a most-used tool in Q7 and optionally the second most-used tool in Q14.

The default selection of tools (given in the multiple-choice selections of Q5, Q7 and Q14) as well as the lists of benefits (Q8 and Q15) and drawbacks (Q11 and Q18) were refined during the piloting phase of the survey of professors [5]; the same lists were used for the survey of students. Indeed, the questionnaire is very similar to the professor's questionnaire, presented at [5]. However, the student questionnaire focused on learning modeling using tools, whereas the professor's questionnaire focused on teaching using tools. The wording of the questions was therefore adjusted, so they made sense for the students.

The right-hand column of Table II indicates the sections of this paper where the data is presented.

RQ1 is largely intended to be answered by Q8 and Q15. RQ2 is intended to be answered by Q11 and Q18. RQ3 is intended to be answered by these same questions, after separately analysing the data about each tool. RQ4 is intended to be answered by Q9.

TABLE II: **Student's Questionnaire**

	Question and Answer Type	**Where Discussed in This Paper**
Q1	**Please indicate whether or not you consent to participate:**	
	{"I am a student who has studied software modeling, and I consent to respond to this survey", "I do not consent, and hence prefer not to participate"}	
Q2	**Where do you live/study (select the country)?**	Section 3.1; Figure 1; Tables I, V
	{Selection from complete list of countries]	
Q3	**Are you currently…?**	Section 3.1; Figure 2
	{"Undergraduate student", "Masters graduate student", "PhD graduate student", "Other"}	
Q4	**Considering specifically learning modeling in courses (but not in thesis research), to what extent have you used modeling tools to support this learning?**	Section 4.1; Figure 3, Table III
	{"Extensively", "Moderately", "A little", "Not at all"}	
Q5	**What tool(s) have you used in support of learning about modeling?**	Section 4.2; Tables IV, V
	{Multiple selection from list of tools given in Table IV or "Other" with ability to supply additional tools in freeform text}	
Q6	**If you chose "other" tool in Question 5, please specify:**	
	{Freeform}	
Q7	**If you chose more than one option in Question 5, which one did you use most?**	Section 4.3; Tables I, VI
	{Single selection from list of tools given in Table IV or "Other"} **If you chose "Other", please specify:** {Freeform}	
Q8	**What are the main benefits of this tool that helped you learn about modeling:**	Section 4.4; Figures 4,5; Tables VII, VIII
	{Multiple selection from list of 17 benefits shown in Figure 4 or Table VII, or "Other"} **If you chose "Other", please specify:** {Freeform}	
Q9	**How difficult was it to use this tool:**	Section 4.6; Figures 7,8
	{"very difficult", "difficult", "neutral", "easy", "very easy"}	
Q10	**Approximately how many hours, did you spend over the last year using this tool in all your classes that taught modeling**	Section 4.8; Figure 10, Table VIII
	{"up to 1 hour", "1-2 hours", "2-5 hours", "5-10 hours", "10-15 hours", "more than 15 hours"}	
Q11	**Select any drawbacks of the tool that bothered you:**	Section 4.5; Figure 6; Table IX
	{Multiple selection from list of 9 drawbacks shown in Figure 6 or Table IX or "Other" } **If you chose "Other", please specify:** {Freeform}	
Q12	**To what extent do you believe this tool influenced your learning of software modeling?**	Section 4.7; Figure 9
	{"a lot", "moderately", "a little", "not at all"}	
Q13	**What are the biggest improvements you would like to see made to this tool:**	Section 5
	{Freeform}	
Q14	**If there is a second tool do you use a lot in support of learning about modeling, please select?**	(as for Q7)
	Same as Q7	
Q15-Q20	***Repeats Q8 – Q13 for the tool selected in question 14***	(as for Q8-Q13)
Q21	**Thank-you for responding to the survey. If you would like to share any additional comments or experiences about learning modeling using tools, please enter them below:** {Freeform}	Section 5

4. Results

In this section we present the results of the survey. We start in Section 4.1 by looking at the extent to which students report using modeling tools to support their learning. In Section 4.2 we discuss the complete set of tools the students report using for modeling, and in Section 4.3 we discuss the tools the students report using most. In Section 4.4 we discuss the benefits the students perceive to be provided by each of these most-used tools, while in Section 4.5 we discuss the perceived drawbacks. Subsequent subsections discuss how difficult the tools are to use, to what extent the tools influenced the students' learning of modeling, and the number of hours students used the tools. In several places we separately analyse results for different geographical areas, as well as for subsets of students who have used a given tool heavily or more lightly.

4.1 To what extent have the students used tools to support learning modeling?

The majority of the 118[2] students who responded to Q4 of the survey reported that they use modeling tools moderately (52.3%) to support their learning of modeling (Figure 3). Only 21% of the students have used tools extensively. This might be due to weaknesses of the tools, or the choice of tools and educational approaches selected by professors.

In Figure 3, we compare this data to that obtained from the professor's survey [5]. The main difference is that more professors claim they use tools extensively than claimed by the students. This difference is small enough, however, to be explainable by sampling bias, or simply by the fact that professors may be using them for tasks other than teaching.

We converted the data from Q4 into a numeric scale where 'Not at all' was counted as 0, and 'Extensively' was counted as 3. The mean was 1.9 (just below 'Moderately') and the standard deviation was 0.7.

We further analysed the data by comparing the responses of students from the four most-represented regions: Canada, the USA, Brazil and the European Union (EU). For this analysis we left out the two Chinese students and the one student who did not identify their country.

Table III shows that the means were very similar, although the European students sampled seem to believe the tools have supported their learning a little more than students in the other

[2] In much of the analysis later in the paper, the count of students is 117. That is because one of the students did not provide their country of origin and certain other data, so we could not use that student's data in all analysis.

regions. Comparing pairs of means, using T tests, with Bonferroni correction [17] (to counteract the likelihood of obtaining significant results erroneously when applying a large number of tests) the only significant difference was between Brazil and the EU (p = .003).

TABLE III: **Regional responses to Q4 regarding extent to which students used tools to support learning of modeling (0 = not at all; 1 = a little; 2 = moderately; 3 = extensively)**

Region	n	Mean	St. Dev.
Canada	35	1.94	0.73
USA	15	1.87	0.83
Brazil	36	1.72	0.66
EU	29	2.17	0.60

The most important lesson from this data is that students appear to be using tools sufficiently to gain experience with them, and hence be in a position to give useful evaluations of their experiences.

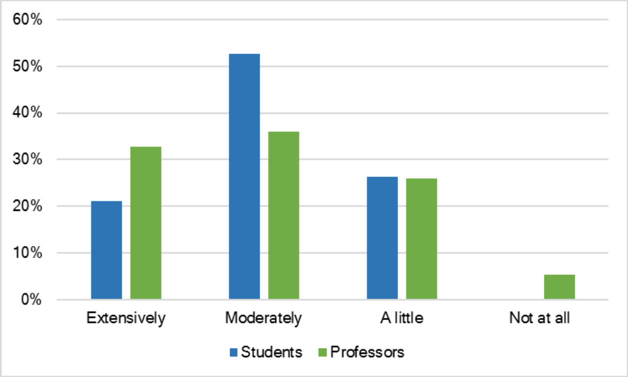


Figure 3. The extent of use of tools to support modeling learning, from Q4 of the survey

4.2 Tools used in modeling learning

Table IV presents *all* 31 modeling tools reported to have ever been used by the 117 students (Q5 and Q6), ranked by percentage of students that have used the tool. More information about each tool can be found at the URL listed in column 3 of Table IV. For tools described in the academic literature, we have provided selected references.

Tools marked with * in column 4 are those that were identified in the pilot study in our professors' survey [5] as the most commonly used tools, and hence were included in the multiple choice selections of Q5, Q7 and Q14. The other rows in Table IV were entered free-form in Q6 by the students after they answered ‘Other’ in Q5. Etrice and MetaEdit also appeared in the multiple-choice list, but were not used by any of the students in the survey.

Tools highlighted in bold are the ones where we received sufficient detailed responses in questions 7-20, such that we were able to perform analysis presented in Sections 4.3 to 4.8.

As a reminder, the ranking cannot be generalized to the population, since the sample was not sufficiently random. We do not intend to claim that Papyrus is the tool that more students have used than any other. However, the ranking provides useful information to help us interpret the remainder of the data because it tells us that students had considerable experience with a wide variety of tools, particularly the ones near the top of this list, and therefore their judgments about benefits and drawbacks are likely to themselves be generalizable. The key message of this paper is about the benefits and drawbacks, not the ranking of tools.

TABLE IV: **The complete list of tools reported to have *ever* been used by the students in the survey (from Q5 and Q6)**

	Tool	%	URL	In Multiple Choice?
1	**Papyrus** [18] [19]	29.1	eclipse.org/papyrus/	*
2	**StarUML**	29.1	staruml.io/	*
3	**ArgoUML**	23.1	argouml.tigris.org/	*
4	**Umple** [10] [20]	22.2	www.umple.org/	*
5	**Magic Draw UML**	20.5	www.nomagic.com/	*
6	**USE** [21]	16.2	sourceforge.net/projects/useocl/	
7	**Astah**	12.8	astah.net/	*
8	**Visual Paradigm**	12.8	www.visual-paradigm.com/	*
9	**Eclipse Mod. Tools**	9.4	eclipse.org/modeling/	
10	Visio	6.8	visio.microsoft.com/	
11	Drawio	5.1	www.draw.io/	
12	yUML	4.3	yuml.me/	*
13	IBM RSM	4.3	ibm.com/software/rational/	*
14	IBM Rat. Rhapsody	4.3	ibm.com/software/rational/	*
15	Acceleo	3.4	eclipse.org/acceleo/	*
16	Visual Studio	3.4	www.visualstudio.com/	
17	PlantUML	2.6	plantuml.com/	*
18	Enterprise Architect	2.6	www.sparxsystems.com/	
19	Rational Rose	2.6	ibm.com/software/rational/	
20	jUCMNav [22]	2.6	jucmnav.softwareengineering.ca/	
21	BoUML	1.7	www.bouml.fr/	*
22	TouchCORE [23]	1.7	touchcore.cs.mcgill.ca/	
23	Dia	1.7	dia-installer.de/	
24	LucidChart	1.7	www.lucidchart.com/	
25	yED	0.9	yworks.com/products/yed/	
26	Violet	0.9	alexdp.free.fr/violetumleditor/	
27	OmniGraffle	0.9	omnigroup.com/omnigraffle/	
28	OSATE	0.9	osate.org/	
29	Cacoo	0.9	cacoo.com/	
30	Edraw	0.9	www.edrawsoft.com/	
31	Gliffy	0.9	www.gliffy.com/	

Table V presents distribution of tools that were reported to have ever been used, combining data from students and professors [5], and ordered by the number of countries where the tool was used. Only tools used in more than one country are shown. The tools with the greatest coverage in different countries/regions were: ArgoUML, Magic Draw, IBM Rational Software Modeler (RSM), Papyrus, Visual Paradigm and StarUML.

TABLE V: **Country distribution of ever-used tools, based on combined responses of students (Q5 and Q6 of our survey) and the same question asked of professors in [5]**

Tools	Number of countries	Country Distribution
ArgoUML	15	Austria, Brazil, Canada, China, France, Germany, Italy, Peru, Poland, Portugal, Saudi Arabia, Sweden, UK, Uruguay, USA
Magic Draw UML	14	Austria, Brazil, Canada, France, Germany, Hungary, Italy, Nigeria, Singapore, Spain, Sweden, UK, Uruguay, USA
IBM Rational Software Modeler (RSM)	13	Argentina, Austria, Brazil, Canada, China, France, Germany, Hungary, New Zealand, Nigeria, Peru, Poland, Saudi Arabia, Spain
Papyrus	13	Austria, Brazil, Canada, Denmark, France, Germany, Hungary, Italy, Spain, Sweden, UK, Uruguay, USA
Visual Paradigm	13	Austria, Brazil, Canada, China, Germany, Israel, Italy, Nigeria, Peru, Poland, Spain, Uruguay, USA
StarUML	12	Brazil, Canada, Germany, Italy, Nigeria, Peru, Poland
Acceleo	11	Brazil, Canada
Eclipse Modeling Tools	9	Brazil, Canada, China, Denmark, Germany, Italy, Poland, Spain
Astah	6	Brazil, Singapore, Spain, UK, Uruguay, USA
IBM Rational Rhapsody	6	Austria, Brazil, Canada, China, Italy, USA
Umple	5	Canada, USA, Italy, France, UK
Draw.io	4	Brazil, Canada, Uruguay, USA
PlantUML	3	Brazil, Canada, France
Visio	3	Canada, China, USA
yUML	3	Canada, Denmark, USA
Cacoo	2	Brazil, Spain
jUCMNav	2	Canada, USA
USE	2	Germany, Spain

4.3 The most heavily-used tools

The tools that are the most heavily used by the participants to learn about modeling are presented in Table VI. By heavily used, we mean tools that students reported to have used either most (Q7) or second-most (Q14), and for which they provided a detailed response regarding its strengths and weaknesses. Table VI lists the tools ranked by the number of detailed responses received (column 3). Not shown are the following tools, where we received just a single detailed response: Acceleo, yUML, yED, Violet, OmniGraffle, Cacoo, Dia, and LucidChart.

Column 4 of Table VI shows the percentage of students who ranked this as their most-used tool (Q7), which sums to 100%, other than rounding. Where any of these tools is a second-most-used tool (Q14), that is accounted for in column 5; only 33% of students gave us data about a second-most-used tool.

TABLE VI: **Most heavily used tools among the participants, ranked by amount of detailed responses we received (data is from Q7 and Q14). Bold tools are analysed further in this paper. T in column 1 means tie.**

RANK	TOOL	# detailed responses	Most-used tool	Second-most used tool	Free & open source	Key features (from their website)
1	**StarUML**	**22**	**15.8%**	**3.5%**	**Only old 2010 version**	**Full UML2 + ERD, DFD; cross-platform; used extensively in industry; model analysis; some code generation; actively maintained; some open-source plugins**
2	**Umple**	**20**	**11.4%**	**6.1%**	**Yes**	**Textual and diagrammatic modeling; UML2 subset (class, state) + ERD; blends model and code; cross-platform; model analysis; code generation; multiple-IDE (command-line, Eclipse, Web); actively maintained**
3	**Magic Draw**	**19**	**16.7%**	**0.0%**	**No**	**Full UML2 + more; cross-platform; used widely in industry; model analysis; code generation; actively maintained;**
T4	**ArgoUML**	**12**	**8.8%**	**1.8%**	**Yes (old)**	**UML 1.x subset; cross-platform; limited code generation; last update 2014**
T4	**Astah**	**12**	**10.5%**	**0.0%**	**No**	**Full UMLs; cross-platform; used widely in industry; some model analysis; code generation; actively maintained**
T4	**USE**	**12**	**0.9%**	**9.6%**	**Yes**	**Textual modeling, with focus on class diagrams (generated) and OCL constraints; cross-platform; specialized model analysis; actively maintained.**
7	**Eclipse Modeling Tools**	**9**	**7.9%**	**0.9%**	**Yes**	**Diverse set of tools supporting full UML2; cross-platform; model analysis; code generation; actively maintained**
T8	**Papyrus**	**8**	**2.6%**	**4.4%**	**Yes**	**Flagship Eclipse project supporting full UML2 and much more; used and actively developed by industrial consortium, cross platform; extensive model analysis; code generation; actively maintained**
T8	**Visual Paradigm**	**8**	**5.3%**	**1.8%**	**No**	**Full UML2 and more; cross platform; model analysis; code generation; actively maintained**
T10	draw.io	5	4.4%	0.0%	Yes	Web based drawing tool with UML notation available but no analysis or code generation; actively maintained
T10	Visio	5	3.5%	0.9%	No	Microsoft office tool with UML notation support, but no analysis or code generation; Windows-only; actively maintained
T12	IBM RSM	3	2.6%	0.0%	No (free download)	UML 2.1; Windows only; mostly superceded by Rational Rhapsody in industry; code generation; not updated since 2015
T12	jUCMNav	3	0.9%	1.8%	Yes	Focus on Use Case Maps and GRL modeling; cross platform; recently maintained
T14	PlantUML	2	0.9%	0.9%	Yes	Textual diagram description for UML (no semantics) and other notations with online diagram generation; actively maintained

It is important to remember that the number of detailed responses we received (column 3) should not be interpreted to reflect the worldwide frequency of use of the tool. It simply helps us judge the reliability of the data for that tool.

Although participants gave detailed responses about each of these 22 tools, we analyse in detail only the nine tools marked in bold in Table IV, for which we received 8 or more detailed responses. We chose this cut-off so as to ensure the reliability of our analysis, and also because the two tools immediately below it are primarily drawing tools and not full modeling tools.

4.4 Perceived benefits of the tools

Question 8 (and question 15 for the second tool) asked, "*What are the main benefits of this tool that helped you learn about modeling*". In this section, we present the results of that question. This responds to our research questions RQ1 (what are the benefits) and RQ3 (which tools have these benefits), as discussed in Section 3. The list of benefits had been developed during pilot testing of our survey of professors. Students were free to select as many or as few of the listed benefits as they wished. If some benefit *mattered* to them in the context of helping them learn about modeling, they would be expected to select it.

As Figure 4 shows, the greatest benefits highlighted in this survey were: 1) being simple to learn; 2) simple to install; 3) free; and 4) supporting the most important modeling notations – all four stood well above other benefits in terms of how students feel they helped them learn to model. Supporting code generation, interacting with other tools, being actively maintained, and supporting textual modeling came next.

The differences between the results from students (top/blue bars in Figure 4 and from professors (bottom/green bars) [5] are relatively small, with the two surveys highly correlating with r=.96. We conducted Z-tests to compare the proportions of students or professors selecting each benefit. More information about Z-test can be found at [25]. The only significant differences were that professors cared more about support for the most important notations (62% vs 48%, p=.03); students care more about the ability to edit models textually (21% vs. 7%, p<.01), and students care about the tool running on the web (14% vs 4%, p<.01). Students also appear relatively more interested in code generation, tool interaction, model-transformation and validation than professors, but these numbers weren't statistically significant.

Being free, supporting code generation, being open source and several other benefits students could choose are objective facts that are either true or false, irrespective of what students select.

However, we wanted to calibrate these benefits among the list of more subjective benefits such as whether or not the tool is simple to use. In other words, we wanted to find out the extent to which being free and other objective qualities) is more or less important than the subjective qualities among users of each tool. The responses must be interpreted in this light.

It is notable that the top three motivating benefits are non-functional attributes (relating to quality or availability). The top functional attributes (involving implemented features) are supporting notations, supporting code generation, interacting with other tools, and supporting textual editing.

Some of the benefits selected least were also non-functional: Students don’t care much about technical support or that the tool is used in industry. They also don’t care particularly for low-rated functional attributes such as “Enables feedback from the professor” and “Standards compliant”.

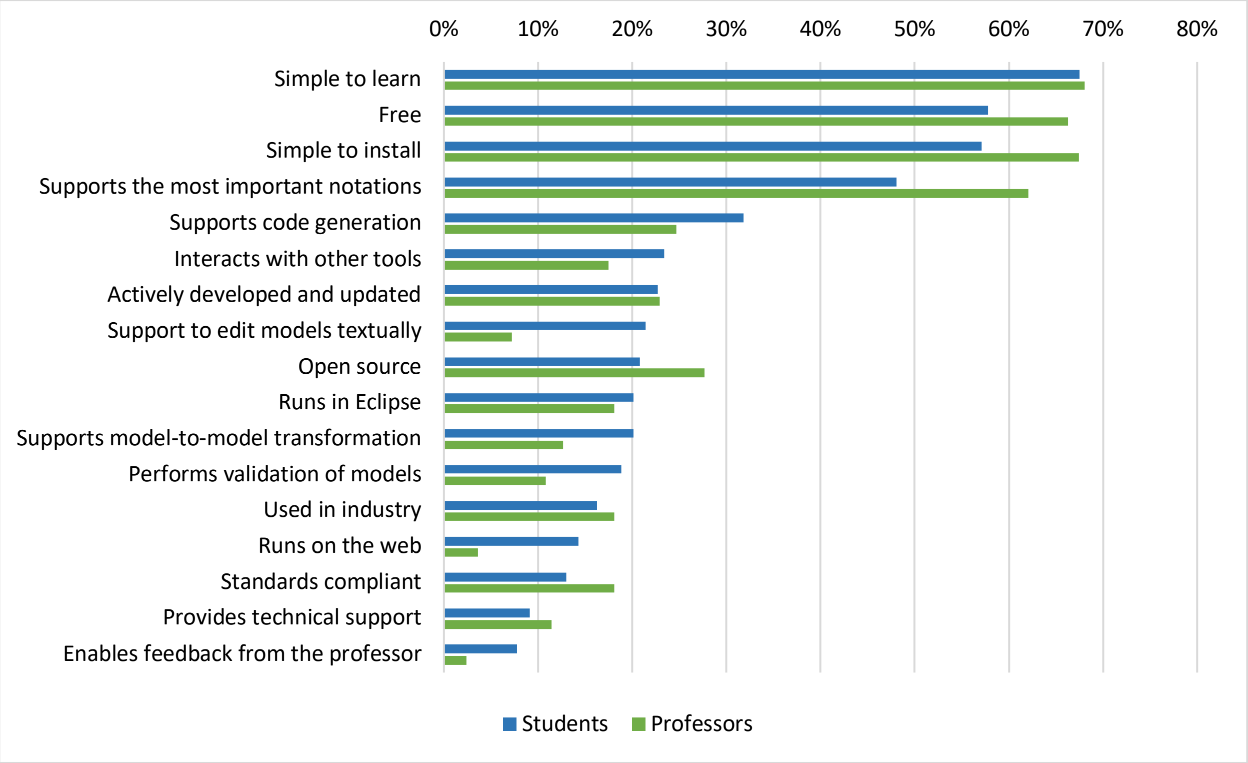
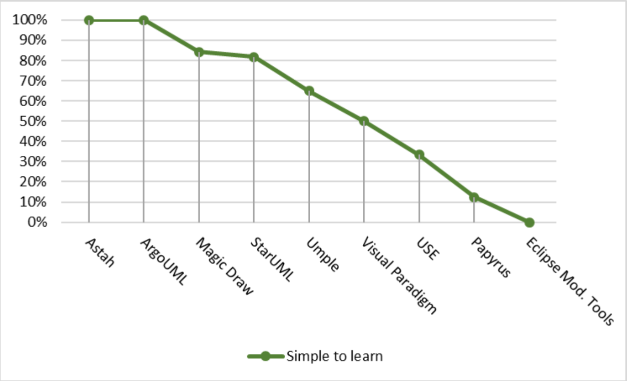


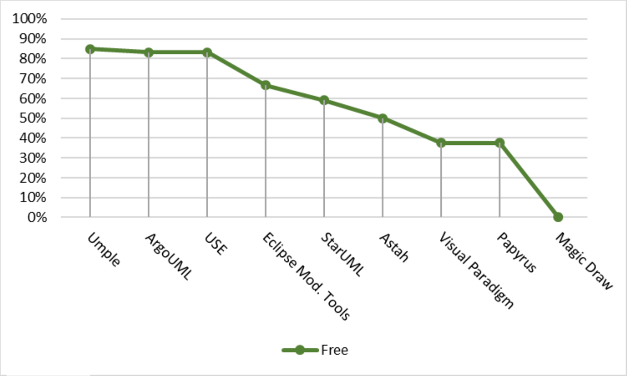
Figure 4.The benefits of the tools that students feel most help them in their learning of modeling from Q8 and Q15 of our survey, compared to data from our earlier survey of professors [5]

Figure 5 (a-e) presents more details about the top five perceived benefits listed in Figure 4. It gives the ratings for each of the nine tools for which we received the most detailed responses.

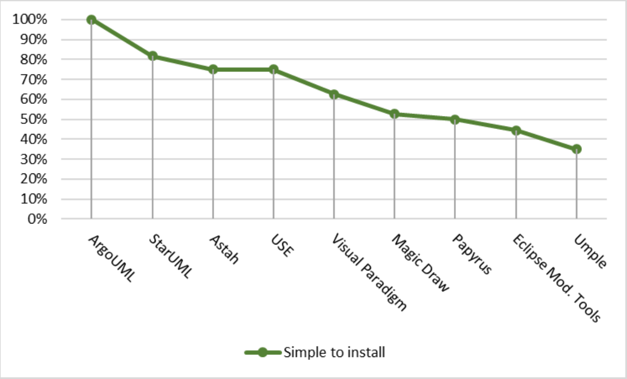
ArgoUML achieved top-three results for most of the benefits, with the exception that it does not offer good support for code generation. Astah also had top-three results in most benefits, except that it is not free. Magic Draw obtained top-three results in being simple to learn, supporting important notations and supporting code generation. Umple obtained top-three ratings in supporting code generation and being free. The only other tool with a top-three rating was StarUML, which was perceived as simple to install. It is interesting to note tools receiving heavy investment, such as Papyrus, did not achieve any top-three results in this analysis.



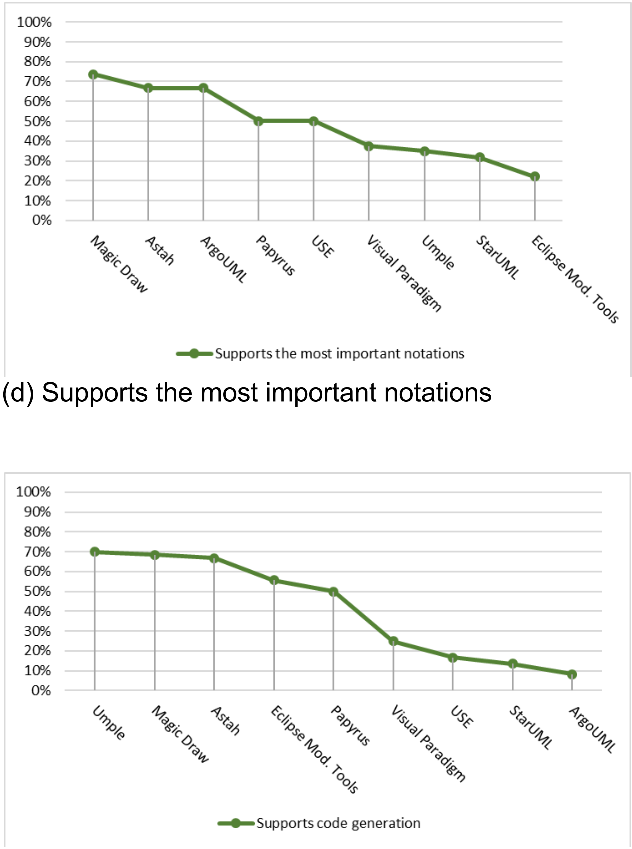
(a) Simplicity of learning



(b) Being free



(c) Simplicity of installation



(d) Supports the most important notations

(e) Supports code generation

Figure 5. Main benefits of the top nine tools from Q8 and Q15 of our survey

Table VII presents the full set of benefits that motivated participants to choose each tool in a different graphical form. The green cells indicate the key benefits, while red and yellow cells suggest areas that were not considered in the choice of each tool. Red and yellow rows may indicate a weakness of the tool or a factor that was not decisive in the choice of the tool. Green cells indicate that a tool has the feature, likely implements it well, and also is something the students are interested in.

We chose the thresholds for the different colours so as to roughly equalize the number of cells with the five chosen colours, yet with the thresholds set as multiples of ten, and with green colours only used when 50% or more of the responses indicated the particular benefit.

By looking for green cells one can see that “Simple to learn” was an important benefit cited by users of the following tools: ArgoUML, Astah, MagicDraw, StarUML, and Umple. “Supports the most important notations” was a strong motivation for most participants in choosing: Magic Draw, ArgoUML, Astah, Papyrus, and USE.

Being free is an interesting case: All the tools marked in green are free, but so is Papyrus which was rated low for this quality. This could be a sampling artifact; it could also be that students are not particularly satisfied with Papyrus overall and didn’t feel like selecting very many benefits (it has no green cells), or that the only benefits they really care about are its code generation and coverage of many notations.

Eclipse Modeling Tools, Magic Draw, Papyrus and Umple present the greatest perceived benefits in terms of code generation. Simplicity to install was an important perceived benefit of ArgoUML, Astah, StarUML, USE and Visual Paradigm. USE stands out from all other tools for users motivated by the benefits “Performs validation of models” and “Support to edit models textually”. Umple had the highest rating for “Runs on the web”, although it was only perceived as a benefit for 50% of its users.

It seems that the students do not find “Enables feedback from the professor”, “Standards compliant” or “Provides technical support” to be qualities that help them learn. However, perhaps tools could still be improved in these areas, since these factors will rate low in this survey if a tool either lacks them, or implements them poorly.

TABLE VII: **Main benefits motivating use of the top nine tools (percentages of responses to Q8 and Q15 received from among students who rated the each tool as their first or second most used tool)**

Tools	Argo UML	Astah	Eclipse M. Tools	Magic Draw	Papyrus	Star UML	Umple	USE	Visual Parad.
Simple to install	100.0%	75.0%	40.0%	52.6%	50.0%	81.8%	35.0%	75.0%	62.5%
Simple to learn	100.0%	100.0%	10.0%	84.2%	12.5%	81.8%	65.0%	33.3%	50.0%
Free	83.3%	50.0%	70.0%	0.0%	37.5%	59.1%	85.0%	83.3%	37.5%
Supports the most important notations	66.7%	66.7%	30.0%	73.7%	50.0%	31.8%	35.0%	50.0%	37.5%
Supports code generation	8.3%	8.3%	60.0%	68.4%	50.0%	13.6%	70.0%	16.7%	25.0%
Actively developed and updated	8.3%	16.7%	50.0%	36.8%	37.5%	13.6%	15.0%	0.0%	25.0%
Interacts with other tools	25.0%	25.0%	20.0%	26.3%	12.5%	27.3%	20.0%	0.0%	25.0%
Supports M2M transformation	25.0%	8.3%	40.0%	26.3%	37.5%	22.7%	10.0%	8.3%	25.0%
Open source	41.7%	8.3%	60.0%	0.0%	0.0%	18.2%	20.0%	16.7%	12.5%
Support to edit models textually	8.3%	0.0%	10.0%	21.1%	0.0%	9.1%	35.0%	58.3%	12.5%
Runs on the web	8.3%	0.0%	0.0%	0.0%	37.5%	13.6%	50.0%	0.0%	0.0%
Standards compliant	16.7%	33.3%	0.0%	10.5%	0.0%	4.5%	10.0%	8.3%	25.0%
Used in industry	16.7%	16.7%	10.0%	42.1%	0.0%	0.0%	10.0%	8.3%	12.5%
Runs in Eclipse	8.3%	0.0%	90.0%	0.0%	37.5%	13.6%	30.0%	8.3%	0.0%
Provides technical support	0.0%	8.3%	10.0%	21.1%	25.0%	18.2%	0.0%	8.3%	0.0%
Performs validation of models	8.3%	0.0%	40.0%	15.8%	25.0%	9.1%	25.0%	66.7%	0.0%
Enables feedback from the professor	0.0%	0.0%	20.0%	0.0%	0.0%	4.5%	20.0%	8.3%	0.0%

🟩	more than 70%
▧	more than 50% and up to 70%
⬜	more than 10% and up to 50%
🟨	more than 0% and up to 10%
🟥	0%

We analysed the data in Table VII to see if there were any strong correlations between the benefits, and conclude that there are not; in other words, the benefits are reasonably independent of each other. Even, for example, the benefit of being free only correlated with being open source moderately (r=.62).

Similarly, we tested to see if there were correlations between the responses given for the various tools: We found that responses for ArgoUML correlated reasonably strongly with responses to StarUML (r=.93), Astah (r=.91) and Visual Paradigm (r=.86). Responses for StarUML also correlated with those for Astah (r=.86) and Visual Paradigm (r=.88). Finally Visual

Paradigm and StarUML responses correlated (r=.83). This suggests that these tools have similar properties, and that the other tools in the list are somewhat distinct from them and from each other. A professor selecting multiple tools, might hence want to just select one from the correlated set {StarUML, Astah, ArgoUML, Visual Paradigm}, plus one of the tools from outside the set.

We analysed responses to Q8 and Q15 to see if there were any differences regarding perceived benefits in the four geographical regions that each had large amounts of data (Canada, USA, Brazil and the European Union). We used Z-tests with Bonferroni correction. The only statistically significant differences were as follows: Brazilian students cared more than Canadians about a tool being easy to install (73% vs. 39%; p<.01), and about being simple to learn (86% vs. 57%; p<.01). Canadian students care more about a tool running on the web than EU students (27% vs. 0%; p<.01), and USA students also care more than EU students about this (28% vs. 0%; p<.01). These differences mostly relate to the tools commonly used in the regions, so this information may not be particularly useful.

We also analysed responses to Q8 and Q15 by comparing cases where students have modeled *heavily* with a given tool (>= 15 hours, as per Q10 and Q18 discussed in Section 4.8) as opposed to cases where students have used the tool lightly (<15 hours). A total of 48 or 31% of the tool uses (most-used or second-most used) were reported as having been used for more than 15 hours (i.e. heavily), and the remainder were reported as being lightly used. The responses regarding benefits between these two groups were reasonably highly correlated (r=.78), but there were some notable differences: Table VIII shows the differences between these groups; statistically significant differences are marked with an asterisk: The vast majority of benefits were more important to heavy tool use than to light tool use. Heavy use resulted in users being more than twice as likely to consider that support for code generation, interaction with other tools, feedback, standards compliance and validation are key benefits. Such users also were more than twice as concerned about whether a tool is being actively developed. The only benefit that light users reported to be noticeably more important to them than heavy users, was the tool being free, although statistical significance wasn't quite reached (p=.06). That makes perfect sense, since a light user won't get as much value from their payment as a heavy user.

This data could be useful to a professor choosing a tool, because they might choose a different tool if they expect students to use it lightly, rather than heavily.

TABLE VIII: **Benefits of the tools indicated in Q8 and Q15 as perceived by heavy users (>=15h) vs. light users (<15h) as per Q10 and Q18. * means the columns are statistically significantly difference (p<.05) according to Z-tests. Underlighted means the difference is more than double.**

Benefit	Importance % to Heavy Users (>= 15 hours of use)	Importance % to Light Users (< 15 hours of use)
Simple to install	54.2	57.9
Simple to learn	**70.8**	64.5
Provides technical support	**14.6**	6.5
Has a way for the user to enter and edit their complete model textually	20.8	21.5
Supports model-to-model transformation	20.8	20.6
Supports code generation (model-to-code transformation)	**54.2 ***	22.4
Interacts with other tools (e.g. for import/export)	**35.4 ***	18.7
Enables feedback from the professor or teaching assistant	**16.7 ***	4.7
Supports the most important notations (e.g. UML diagrams) you were learning about	**60.4 ***	42.1
Actively developed and updated	**41.7 ***	13.1
Free	45.8	**62.6**
Open source	18.8	21.5
Runs on the web	12.5	15.9
Runs in Eclipse	**27.1**	16.8
Standards compliant	**22.9 ***	8.4
Performs validation of models	**31.3 ***	14.0
Used in industry, so as to prepare you for industrial practice	**25.0**	13.1

4.5 Drawbacks of the tools

Figure 6 summarizes the data from Q11 and Q18, regarding perceived drawbacks of the nine most heavily used tools. This responds to our research questions RQ2 (what drawbacks are perceived) and RQ3 (which tools have those drawbacks), as discussed in Section 3. The specific question asked the students to, “select any drawbacks of the tool that bothered you”. The list of drawbacks was determined in the pilot of the survey of professors [5], although students were able to add their own ‘other’ choices.

The biggest drawbacks were, “Does not give feedback about models”, “Slow to use” and “Hard to draw diagrams that appear the way I would want”.

Unlike the benefits discussed in the last section, the results for the drawbacks were significantly different from the results of the professors’ survey [5], which are shown here in green. In this topic, the correlation between the two surveys was low, r=.18. Not giving feedback, being slow to use, not drawing diagrams the way the student wants, and being buggy were all statistically significantly more important to students (p<.01) than to professors. On the other hand, not interacting well with other tools was statistically significantly more important to professors than to students. It might be that professors are not aware of some of the user experience issues that students encounter.

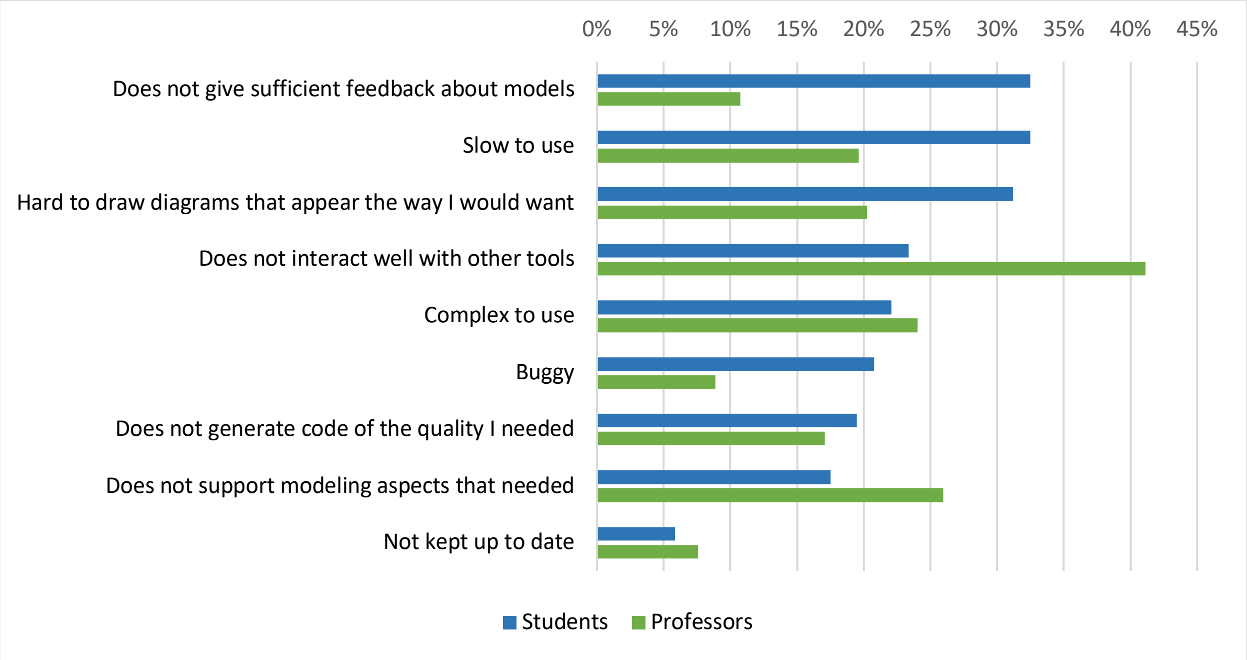


Figure 6. Drawbacks of the most-used and second most-used tools, from Q11 and Q18 of the survey

Table IX shows the drawbacks related to the nine most-heavily-used tools. In this table, the green cells indicate areas where participants are generally satisfied, while the red cells suggest areas of general dissatisfaction or feature gaps.

We chose the thresholds for the colours using the same rationale as for Table VII, except that now low numbers mean 'good'.

In the opinion of the students, Eclipse Modeling Tools are the most complex among the top nine tools, followed by USE. Umple is rated as the least complex.

ArgoUML and Astah have the highest percentage of complaints about lack of feedback. Several other tools also have this weakness. Papyrus, Umple and USE are reported to give good feedback. ArgoUML, StarUML and USE seem to give students the least ability to draw diagrams the way they want.

Umple and Papyrus were rated as the most buggy at the time of the survey. We did not ask about the types of bugs and left interpretation of what 'buggy' means to the students, but pointing this out should help prompt the tool developers to put extra effort into solving bugs. These same tools, along with MagicDraw and the Eclipse tools, are flagged as having some speed problems.

None of the top nine tools resulted in any concerns regarding being kept up to date.

It is notable that almost all tools have some red or yellow areas, but all of the tools also have dark green areas. This suggests that there is a lot of diversity among the tools, and all have strengths.

TABLE IX: **Main drawbacks noted among the top tools for which responses were received to Q11 and Q18.**

Tools	Argo UML	Astah	Eclipse M. Tools	Magic Draw	Papyrus	Star UML	Umple	USE	Visual Parad.
Does not give sufficient feedback about models	50.0%	41.7%	40.0%	33.3%	0.0%	31.8%	15.0%	16.7%	25.0%
Hard to draw diagrams the way I want	50.0%	16.7%	20.0%	11.1%	25.0%	45.5%	40.0%	41.7%	25.0%
Slow to use	16.7%	16.7%	50.0%	61.1%	50.0%	22.7%	55.0%	25.0%	25.0%
Does not interact well with other tools	25.0%	25.0%	20.0%	27.8%	25.0%	31.8%	25.0%	16.7%	12.5%
Does not generate code of the quality I needed	25.0%	16.7%	0.0%	16.7%	25.0%	31.8%	15.0%	16.7%	12.5%
Complex to use	16.7%	16.7%	60.0%	16.7%	25.0%	18.2%	5.0%	50.0%	12.5%
Buggy	16.7%	0.0%	20.0%	22.2%	37.5%	22.7%	55.0%	0.0%	12.5%
Does not support modeling aspects that needed	16.7%	16.7%	0.0%	11.1%	12.5%	27.3%	20.0%	25.0%	0.0%
Not kept up to date	8.3%	8.3%	0.0%	5.6%	0.0%	4.5%	15.0%	8.3%	12.5%

🟩	up to 10%
▩	between 10% and 20%
⬜	between 20% and 30%
🟨	between 30% and 40%
🟥	more than 40%

As with the tool benefits, we analysed the drawbacks data to see if there were any geographic differences between Brazil, Canada, the USA and the EU. The only statistically significant difference was that students in the EU were more concerned about complexity than Canadian students (66% vs. 10%, p<.01). Likely this simply reflects the complexity levels of the tools more commonly used in each region.

We also analysed the data to see if there were differences between light tool users (<15 hours) and heavy tool users (> 15 hours): there were not. Correlation between the answers from the groups was high (r=.80) and there were no statistically significant differences. Only one drawback difference almost reached statistical significance: Heavy users were concerned about tools not interacting with other tools, to a greater extent than were light users (33% vs. 20%, p = .06).

4.6 How difficult were the tools for students to use

We asked students about difficulty of use to obtain more general evidence about the tools and answer our research question RQ4. This can be used together with the evidence about benefits and drawbacks discussed in the previous two sections. Most students considered the tool they used most heavily to be simpler and easier to use than the second-most tool they used (Q9 and Q16), as presented in Figure 7. This confirms simplicity as an important aspect to be considered in choosing a tool. This also is evident in Figure 4.

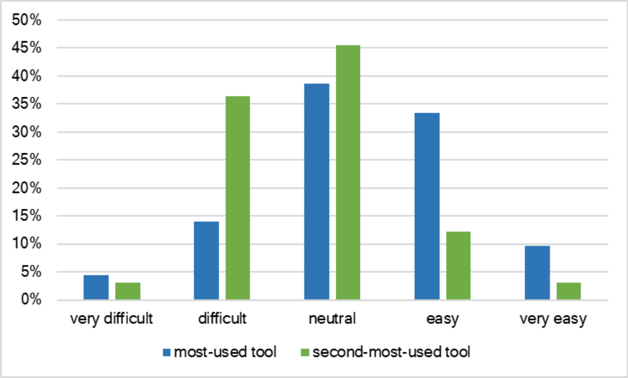


Figure 7. Difficulty for students to use the tool (data from Q9 and Q16)

Figure 8 presents the difficulty level students reported experiencing (Q9 and Q16) in the use of the top nine tools. Astah, ArgoUML, Umple and StarUML were most cited as "very easy" to use. On the other hand, Eclipse Modeling Tools had the most responses indicating "very difficult", followed by Visual Paradigm and Papyrus. Figure 8 is ordered easiest-to-hardest from left to right, by weighting "very easy" as having a value of 4, and "very difficult" as having a value of zero.

Using the above numeric conversion, the mean was 2.2 (or slightly more easy than neutral) and the standard deviation was 1.1.

The data from Q9 and Q16 is roughly consistent with the data from our survey of professors [5]; the most notable exception is that professors judged Visual Paradigm to be much easier to use than the students did. It might be that it requires extensive learning, but once learned is easy to use.

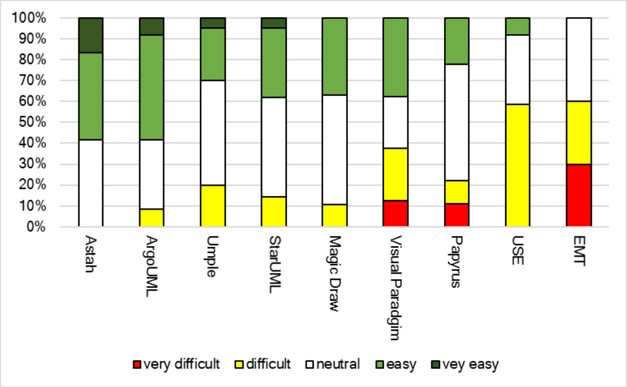


Figure 8. Difficulty level of the Top 9 most-used tools, from Q9 and Q15 of the survey

4.7 To what extent did the students believe the tool influenced their learning of modeling

Most of the participants indicated in their response to Q12 and Q19 that the use of their chosen tools moderately influenced their learning of modeling (Figure 9). However, the use of tools has greater impact on the modeling learning for students than for professors.

Converting the data to a scale where "not at all" is zero and "a lot" is 3, the mean response was 1.8 (or just below "moderately") and the standard deviation was 0.9.

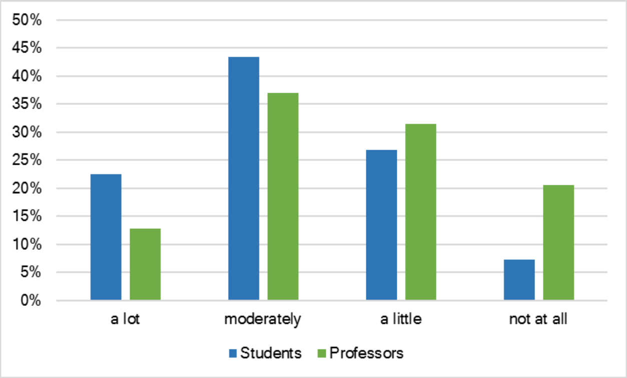


Figure 9. Influence on learning of modeling by using the tools, from the responses to Q12 and Q19

4.8 Hours of use of the tool

Figure 10 is a histogram of the number of hours that students spent, over the last year, using the most-used tools (either their first or second tool) in learning about modeling (Q10 and Q18). Most of the students reported using tools for more than 5 hours, which shows that students make good use of tools to support their learning, and lends confidence to the other data in this survey. It was not possible to derive a mean number of hours or other statistics from this question due to the fact that the time-periods covered are not uniform, and the final one is open-ended, but we did use this data to divide the sample for questions discussed in earlier sections of this paper.

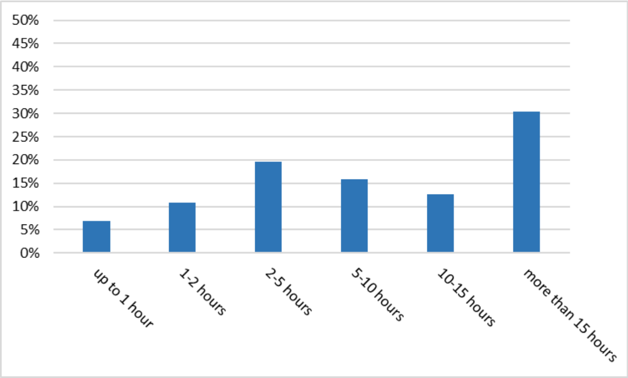


Figure 10. Number of hours of use of each tool in the classroom, from Q10 and Q18

5. General Discussion

In this section we discuss a few general issues raised by the survey.

Clearly, it is instructors who generally choose most tools that students use, to some extent based on their own experiences with the tools, to some extent based on marketing of tools and use of the tools by colleagues, and to some extent based on the features they want students to use (e.g. code generation, or particular diagram types). The fact that an instructor *chose* a particular tool, however, might be expected to be reasonably independent of the *benefits or problems* students perceived with the tool. An exception would be when feature is missing, and an instructor did not direct the student to use such a feature (e.g. code generation), then students would be less likely to consider it as a disadvantage. If, over time, features such as code generation become more important to the pedagogical method of instructors, then the student responses regarding tools where such features are weak or absent would naturally become more negative.

The tools used by the students varied greatly in capability and intent. Some tools attempt to faithfully present all of standard UML, others present a subset, and still others are simply drawing tools. It would be expected then that in courses attempting to cover UML itself comprehensively, professors would choose tools of the former type, and both students and professors might complain about gaps in the latter type.

People who discussed this survey with us questioned why we included drawing tools such as Visio, since they clearly have important disadvantages. But the set of tools was self-selected by the professors, and some professors, for whatever reason (perhaps perceived simplicity), arrange for students to use drawing tools. It is only fair that those be ‘in the ratings’, even though we clearly expect them to be rated low in several respects.

Support for code generation also varies widely – from none at all to comprehensive. Not all pedagogical methods involve having students generate code, and this likely influences tool choice. There is likely only to be negative or positive judgment about the quality of code generation from students who actually tried to use it.

The survey focused a lot on uncovering weaknesses of tools (lack of features, difficulties, bugginess). Even in the discussion of perceived benefits, it was the lack of such benefits (not simple, not free, not standards compliant, does not run in Eclipse, does not perform validation, etc.) that turned out to be most informative. We believe that this is entirely reasonable: After all, an ideal tool would have all the benefits and none of the drawbacks, and an objective of this work was to help guide users to select tools that are closer to that ideal, and to prompt developers (including the authors of this paper) to keep improving their tools with that ideal in mind.

In questions 13 and 20 we gave students the opportunity to give freeform answers proposing improvements to the tools. Question 21 also asked for additional freeform comments. The vast majority of the suggestions simply reinforced the answers they had already made in selecting the drawbacks in questions 11 and 18, particularly requesting improvements to the User Interface (UI), to response time, and for the removal of bugs. A small number of other requests proved interesting.

In particular, user interface improvements requested for several tools included the ability to access features using fewer menus, or with more shortcuts; the ability to make repetitive tasks faster; easier installation; real-time editing with collaborators; better documentation, and the ability to edit invalid models so the tool can be used for sketching (i.e. disabling forced validation).

In Question 21, a couple of students pointed out that they also used modeling tools outside academic courses, and had restricted their answers to the tools they had used in courses.

6. A Magic Quadrant for modeling tools used in education

We adapted the concept of the Magic Quadrant, described in [24], to evaluate the top nine tools analysed in this research. The result is presented in Figure 11.

In the Magic Quadrant method, the X axis measures 'Completeness of Vision'. A more complete product will have more features and fewer perceived weaknesses. We have derived data for this scale from our tables of benefits (Table VI) and drawbacks (Table VII). Weights were mapped to each color assigned to the benefits and drawbacks, as follows: dark green (2), light green (1), white (0), yellow (-1) and red (-2). The final score was obtained based on the weighted sum of benefits and drawbacks for each tool.

The Y axis measures 'Ability to Execute', or market penetration. For this we have used the ranking of the ever-used tools by professors and students combined.

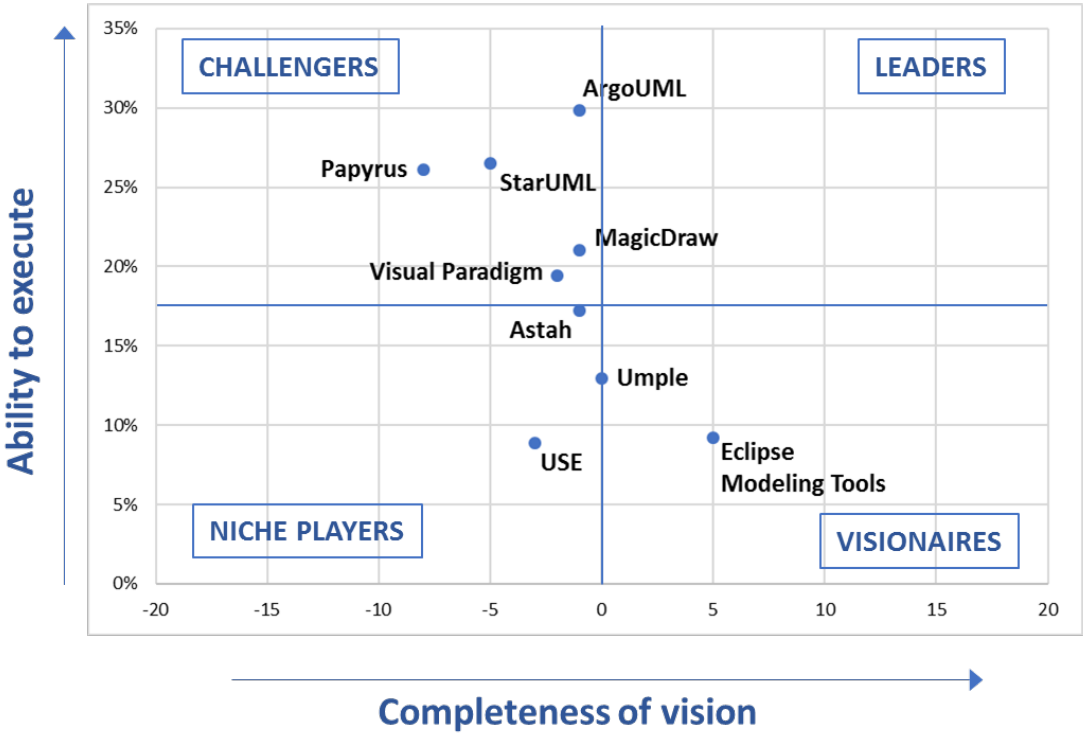


Figure 11. Magic Quadrant – Top nine tools

The Magic Quadrant method results in division of tools into four quadrants.

The **challengers** are those tools that are widely used, but for which there are opportunities for improvement. In our survey, these are: ArgoUML, StarUMlL, Papyrus, Magic Draw and Visual Paradigm.

Visionary tools set the trends and embrace the evolution of technology, but these tools have a relatively weak market share for the market being analyzed (tool use in teaching modeling). The only tool solidly in this category is 'Eclipse Modeling Tools'. Given the widespread use of Eclipse, this may seem surprising, but our survey indicates that at least in education the Eclipse Modeling Tools are not being chosen as frequently as other tools (Note that Papyrus is treated separately in our survey). Umple is on the boundary with the next category.

Niche players tend to focus on a specific segment. Niche players may support specific needs better than other tools, and are not leading market players. Astah and USE fit in this category, with Umple being on the border with the category above.

No tool in our survey currently can be considered a **Leader**, meaning it would have a high market share and *also* a broadly satisfied user base covering all important features.

7. Threats to validity

Earlier in this paper we pointed out one key threat to validity: Although we sampled from a respectable number of students, and from a variety of countries, the number of institutions represented was limited and not fully random, so the data about which tools were the most frequently used is clearly not representative of worldwide educational tool use. For example, although the most-used tool in our survey among ever-used tools was Papyrus with 29.1% of students having used it, readers of this paper cannot conclude that 'Papyrus is used by 29.1% of students', since we have likely over-sampled institutions that happen to prefer Papyrus. The same is true for the 'most used' tools where StarUML comes out on top with 22 of 151 of our detailed responses (the 152 responses include most-used and second-most-used tool).

There could be biases that made certain students keener to complete the survey if they strongly like or strongly dislike certain tools. However, this is a general problem with surveys and we do not believe it is likely to result in strong bias, since the frustrated students are likely to cancel out the ones who love their favourite tool.

Some tools are used only in a single country or predominantly so: MagicDraw, Papyrus and USE in Spain; Astah, ArgoUML and StarUML in Brazil; Umple in Canada, and Eclipse Modeling Tools in Denmark. The question arises as to whether this threatens the analysis. It should only do so if the cultural differences of students in each country have a dominant influence over which benefits and weaknesses they feel most impact them. Since the survey results aligned so much with our survey of professors, we believe this was not a significant issue.

Several of the professors happen to be tool developers or are associated with open source communities developing some of the tools (We are aware of this being the case for Papyrus, USE, and Umple). In particular, one of the authors of this paper leads the Umple project. This might have led to bias if those students want to please their professors by agreeing to participate in the survey. However, several factors mitigate against such a bias: 1) Each of the tools in question received evaluations from multiple universities, not just the home university of the professor involved in development, and the evaluations didn't differ notably; 2) The results were gathered anonymously, such that it was impossible for any of the professors to know which students in fact participated; 3) Each of the tools in question received both positive and negative ratings, and these were consistent with the ratings given in our survey of professors [5].

Another possible threat is construct validity: It is possible that the way we phrased questions might have led to certain conclusions, and that if we had phrased questions in other ways, we might have obtained different results. For example, for the questions regarding benefits, we focused on the benefits that would motivate participants to use this tool. That way of constructing the question did not distinguish between responses based on the presence/absence of features, vs. the feature being well-implemented or poorly implemented in a tool. We could have made the survey considerably longer and more complex to tease out such distinctions, but at a cost of a potentially-lower participation rate: We wanted to be able to tell participants that they survey would only take about 10 minutes.

In various places in the paper we have compared this survey with our earlier survey of professors. A question that might be asked is, "can the data be reasonably compared?" To ensure compatibility, we made a point of asking the *same* questions, adjusted only for words such as 'teach' vs. 'learn': The answers to questions about tool strengths and weaknesses should therefore be comparable. The professors came from many different universities in many more countries than the students. There is some potential that the student results might be more biased towards particular cultures or the nuances of how modeling is taught in those countries. However, we believe the probability of this is low because we did receive results from seven countries and four continents.

Finally, the data about which specific tools have which specific benefits and drawbacks can only be considered valid at the time of the survey. Although some tools have essentially ceased development (e.g. ArgoUML), the data would be less applicable to newer versions of tools that are being actively developed. On the other hand, the general results regarding which benefits and drawbacks are considered important should remain valid as it has been averaged over many tools.

8. Conclusions and future work

In this study we surveyed over 100 students to learn which modeling tools they have used, as well as their perceptions of the benefits and drawbacks of these tools. Our overall objectives were to guide ongoing modeling tool development, and to help both professors and students choose suitable tools.

The most important contributions of this research are the following:

- We have identified a set of benefits that students feel to be most valuable in software modeling tools: Being simple to learn, free, simple to install and supporting the most important notations are at the top of this list (Figure 4). This addresses our RQ1 presented in Section 3.
- We have identified the list of drawbacks that most bother students about modeling tools: Not giving feedback about models, being slow, and difficulty in drawing diagrams are the top items in this list (Figure 6). This addresses our RQ2 presented in Section 3.
- We have identified the relative strengths and weaknesses of a specific set of widely used tools with respect to the lists of strengths (Table VII) and weaknesses (Table IX), and the overall perceptions of difficulty of use of the tools (Figure 8). This addresses our RQ3 and RQ4 as presented in Section 3. Astah, ArgoUML and Umple are perceived as the most easy-to-use tools (although ArgoUML is no-longer being updated); Eclipse Modeling Tools, Papyrus and USE are perceived as the least easy-to-use. This data comes from enough users in enough regions that it should be considered reasonably reliable.
- We developed a Magic Quadrant showing which tools are Leaders, Challengers, Visionaries and Niche Players.

The following are the key messages we believe can be derived from this research by the three groups of stakeholders: professors, students and tool vendors.

When it comes to benefits of tools, the perceptions of students about modeling tools are similar to the perceptions of their professors. Although perhaps not surprising, this indicates that professors likely appreciate some aspects of their students' experience. It also lends confidence that the weaknesses and strengths we have identified are indeed real. On the other hand, professors did not seem to appreciate all the drawbacks students experience with their tools.

There is a great diversity of modeling tools, with contrasting strengths, weaknesses and feature sets. This explains why educators and students tend to use multiple tools, and suggests that they should continue to do so. It also suggests that tool developers ought to try to learn from the weaknesses highlighted regarding their tools, and also from the strengths of tools that compete with theirs. In the months since this survey was taken, we have noticed that new releases of tools are in fact moving to address some of the weaknesses. Indeed one of the motivations for conducting this survey was to obtain data that would allow developers of Umple to improve it; this survey pointed out speed and bugginess to be weaknesses of Umple, both of which have been to some extent addressed in recent months. Papyrus has also put considerable effort into improving usability.

Students and professors seeking a tool, and developers working on tools, clearly need to focus on the most sought-after quality attributes: being simple to learn and simple to install. These attributes dominated positive attributes sought by both professors and students. On the flip side, improving quality by removing key irritants should be important for developers: The key irritants are lack of feedback from the tools (e.g. analysing models to find errors), slowness, inability to draw diagrams that appear as the user wants, inability to interact with other tools, complexity and bugginess. Students and professors selecting tools should steer away from tools that have these problems.

Once quality aspects are perfected, then the next thing to look at is the feature set: Supporting many notations, doing good code generation, interacting well with other tools and supporting a textual notation were considered the most important. Being free and open source is considered very important too, but this may not easily be under the control of developers. Perhaps developers of expensive tools should try to find ways to make their tools free for students – this may encourage them to use the tools in their studies, and later on pay for them when in the workforce.

This paper has focused on studying perceptions of the difficulty of use, weaknesses and motivating features of software modeling tools in current use. As future work, there is a great deal of opportunity to dig into the details, examining in greater precision which features and user interface designs work best for students and their professors. It would also be important to replicate this work after a few years to see what has changed; ideally the replication might attempt to survey an even greater number of students at an even wider number of institutions.

Acknowledgments

We would like to thank all the participants in this survey and the professors that forwarded the survey to their students. Dr. Lethbridge's research is funded in part by NSERC.

References

[1] Paige, R. F., Polack, F. A., Kolovos, D. S., Rose, L. M., Matragkas, N., and Williams, J.R.: Bad modelling teaching practices. In: ACM/IEEE 17th Int. Conf. on Model Driven Engineering Languages and Systems – Educators Symposium, Valencia, Spain (2014)

[2] Akayama, S., Demuth, B., Lethbridge, T.C., Scholz, M., Stevens, P., Stikkolorum, D.R.: Tool use in software modelling education. In: Proceedings of the ACM/IEEE 16th Int. Conf. on Model Driven Engineering Languages and Systems - Educators Symposium, Miami, USA (2013)

[3] Budgen, D., Burn, A.J., Brereton, O.P., Kitchenham, B.A., Pretorius, R..: Empirical evidence about the UML: A systematic literature review. Softw. Pract. Exp., 41(4), 363-392 (2011)

[4] Forward, A., Badreddin, O., Lethbridge, T. C.: Perceptions of software modeling: A survey of software practitioners. In: Proceedings of the 5th Workshop from Code Centric to Model Centric: Evaluating the Effectiveness of MDD (C2M:EEMDD) (2010)

[5] Agner, L.T.W., Lethbridge, T.C.: A Survey of Tool Use in Modeling Education. In: Proceedings of the ACM/IEEE 18th Int. Conf. on Model Driven Engineering Languages and Systems (MODELS), pp. 303-311 (2017)

[6] Reggio, G., Leotta, M., Ricca, F.: Who Knows/Uses What of the UML: A Personal Opinion Survey. Models 2014, Springer, pp. 140-165 (2014)[7] Kuzniarz, L., Martins, L.E.G.: Teaching Model-Driven Software Development: A Pilot Study. In: Proceedings of the ITiCSE Working Group Reports (ITiCSE '16). ACM, New York, NY, USA, pp. 45-56 (2016)

[8] Tekinerdogan, B.: Experience in teaching a graduate course on model-driven software development. Computer Science Education, 21:4, 363-387 (2013)

[9] Paige, R.F., Polack, F.A., Kolovos, D.S., Rose, L.M., Matragkas, N., Williams, J.R.: Bad modelling teaching practices. In: Proceedings of the ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems – Educators Symposium (Educators' Symposium@MODELS) (2014)

[10] Lethbridge, T.C.: Teaching modeling using Umple: Principles for the development of an effective tool. In: Proceedings of the IEEE 27th Conference on Software Engineering Education and Training (CSEE&T), pp. 23-28 (2014)

[11] Lynch, A., Flango, D., Smith, R., Lang, M.: Experiences of using Rational Rose/Visio for UML modeling in an undergraduate software engineering course: A student perspective. Journal of Computing Sciences in Colleges (JCSC) 19(5), 353-356 (2004)

[12] Liebel, G., Heldal, R., Steghöfer, J.P.: Impact of the Use of Industrial Modelling Tools on Modelling Education. In: Proceedings of the IEEE 29th International Conference on Software Engineering Education and Training (CSEET'16), Dallas, TX, pp. 18-27 (2016)

[13] Giraldo, F.D., España, S., Giraldo W.J., Pastor, O.: Modelling language quality evaluation in model-driven information systems engineering: A roadmap. In: IEEE 9th International Conference Research Challenges in Information Science (RCIS), pp. 64-69, IEEE (2015)

[14] Whittle, J., Hutchinson, J., Rouncefield, M.: The State of Practice in Model-Driven Engineering. IEEE Software, 31(3), 79–85 (2014)

[15] Mussbacher, G., Amyot, D., Breu, R., Bruel, J.M., Cheng, B.H., Collet, P., et al.: The relevance of model-driven engineering thirty years from now. In: Proceeding of the 17th International Conference ACM/IEEE - Conference on Model Driven Engineering Languages and Systems (MODELS). LNCS, 8767, 183–200, Springer (2014)

[16] Tomassetti, F., Tiso, A., Ricca, F., Torchiano, M., Reggio, G.: Maturity of software modelling and model driven engineering: A survey in the Italian industry. In: Proceeding of 16th International Conference on Evaluation Assessment in Software Engineering (EASE), pp. 91-100 (2012)

[17] Weisstein, E.W. Bonferroni correction. MathWorld - A Wolfram Web Resource. CRC Press. Available at http://mathworld.wolfram.com/BonferroniCorrection.html. Accessed October 2018.

[18] Gérard, S., Dumoulin, C., Tessier, P., Selic, B.: Papyrus: A UML2 tool for domain-specific language modeling. LNCS, Model-Based Engineering of Embedded Real-Time Systems, 6100, 361-368 (2010)

[19] Lanusse, A., Tanguy, Y., Espinoza, H., Mraidha, C., Gerard, S., Tessier, P., Schnekenburger, R., Dubois, H., Terrier, F.: Papyrus UML: an open source toolset for MDA. In: Proc. of the 5th European Conference on Model-Driven Architecture Foundations and Applications (ECMDA-FA 2009), pp. 1-4 (2009)

[20] Lethbridge, T.C., Abdelzad, V., Husseini Orabi, M., Husseini Orabi, A. Adesina, O.: Merging modeling and programming using Umple. In: Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications (ISoLA 2016). LNCS, 9953, 187-197 (2016)

[21] Gogolla, M., Büttner, F., Richters, M.: USE: A UML-based specification environment for validating UML and OCL. Science of Computer Programming, 69, 27-34 (2007)

[22] Mussbacher, G., Amyot, D.: Goal and scenario modeling, analysis, and transformation with jUCMNav. Software Engineering-Companion Volume, 2009. 31st International Conference on Software Engineering (ICSE), IEEE, pp. 431-432 (2009)

[23] Duran, M.B., Mussbacher, G.: Evaluation of Goal Models in Reuse Hierarchies with Delayed Decisions. IEEE 25th International Requirements Engineering Conference Workshops (REW), IEEE, pp. 6-15 (2017)

[24] Sallam, R.L., Richardson, J., Hagerty, J., Hostmann, B.: Magic quadrant for business intelligence platforms. CT, Gartner Group (2011)

[25] Montgomery, D. C., Runger, G. C., Hubele, N. F. Engineering Statistics, New York, NY, USA:Wiley, 2009.