DYNAMIC GEOMETRY SOFTWARE IN MATHEMATICAL MODELLING: ABOUT THE ROLE OF PROGRAMME-RELATED SELF-EFFICACY AND ATTITUDES TOWARDS LEARNING WITH THE SOFTWARE

Corinna Hertleif

Westfälische Wilhelms-Universität Münster; c.hertleif@uni-muenster.de

Mathematical modelling is a complex process consisting of several steps, which can also be carried out with the use of digital tools. This paper takes a closer look on how students perceive the DGS GeoGebra when learning mathematical modelling, how their confidence in their tool competencies changes when using the software to do modelling, and if the learning outcome concerning modelling competencies is influenced by programme-related self-efficacy or attitudes towards learning with the digital tool. Results from both qualitative and quantitative evaluations of a study with approx. 300 grade 9 students are reported.

Keywords: DGS, Modelling, Self-efficacy, Attitudes

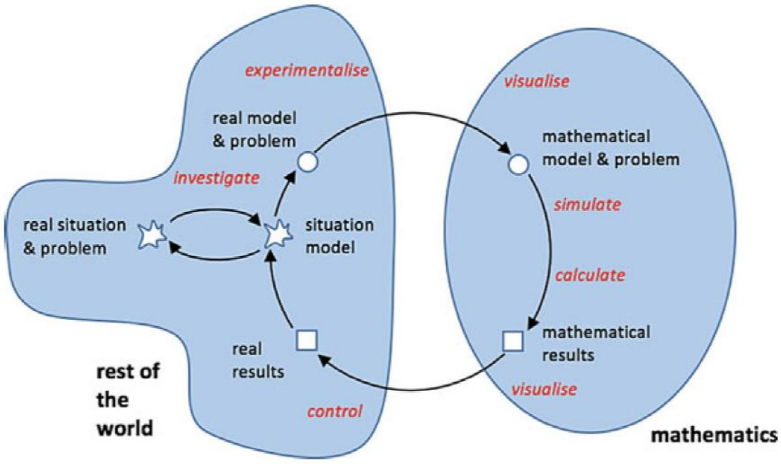
THEORETICAL BACKGROUND

Modelling with Dynamic Geometry Software

Mathematical modelling is a complex process in which a problem in a real-world situation must be understood and simplified and then translated into the world of mathematics to be solved by mathematical means. The found mathematical results then must be related back to the real-world problem and finally be reflected and checked for plausibility. If this check indicates that the found results do not yet represent a satisfying solution, the steps of the modelling process can or should be repeated until a satisfying solution is found (Blum 2015). That is why this process often is displayed as a cycle (see Figure 1), even though the real process of solving a modelling problem does not necessarily have to strictly follow this cycle (Borromeo-Ferri 2006). Modelling consists thus of different steps, most prominently among them are simplifying, mathematising, interpreting and validating. Being competent in modelling therefore means, in a comprehensive sense, being able to construct and to use or apply mathematical models by carrying out appropriate steps as well as to analyse or to compare given models (Blum et al. 2007). The abilities to carry out a certain step of the modelling process respectively are also called sub-competencies of modelling (Blum 2015).

It is also possible to make use of digital tools while modelling. Depending on the kind of modelling problems, spreadsheets, computer-algebra-systems or dynamic geometry software (DGS) may not only support or take on the mathematical work but also visualise models, simulate real-world processes or be used to control mathematical results (Siller & Greefrath 2010). When modelling with a DGS, the software can for example be used to draw or construct geometric models or to measure specific quantities needed to solve a problem. The dynamics of the software are especially useful for a flexible adaptation of already constructed models either with the aim of simulating possible solutions or of improving the used model. Thus, not only the step of mathematising may benefit from the use of a digital tool. Steps like validating or reflecting, which require the adaptation or adjustment of mathematical models, may profit from the use of a digital tool as well (Greefrath, Siller & Weitendorf 2011).

Fig. 1. Possible use of digital tools for modelling (Greefrath et al. 2011)



When using a DGS, the different actions that can be carried out, can be classified into a scheme of operations (Mackrell 2011, Sedig & Sumner 2006). The most obvious actions done with a DGS are probably those in which a new object is constructed using both already existing objects and the available DGS-tools. These construction-operations are supplemented by object-operations in which no new object is created, but existing objects are changed. For example with the help of a drag-mode, it is possible to rearrange objects, to simulate dynamic processes or to vary different parameters. Both construction- and object-operations can enhance the modelling process: in the step of mathematising or validating for example, geometric models can be build using construction-operations and be reflected using object-operations. Additionally, view-operations, in which a constructed object is displayed in an alternative way, e.g. from a different angle, can support reflections regarding the model fit.

But even though a DGS might be supportive for different modelling steps, mathematical modelling remains a cognitively demanding activity, which requires not only mathematical knowledge but also concept ideas, appropriate beliefs, attitudes and extra-mathematical knowledge (Blum 2015). Modelling with digital tools additionally requires skills in certain software tools (Siller & Greefrath 2010). When students are learning modelling with a digital tool, it is possible that two learning processes take place at the same time. On the one hand students have to learn how to deal with complex tasks like modelling problems and on the other hand they have to cope with software to which they might not yet be fully accustomed. Little is known on how students perceive the used instrument when learning modelling with a digital tool, what difficulties they encounter and what strategies they pursue to take full advantage of the instrument's power.

Programme-Related Self-Efficacy and Attitudes Towards DGS

It is known from the Social Cognitive Theory that self-efficacy (SEF), which is the belief that one has the ability to perform a particular action, has a strong influence on behaviour (Bandura 2012). When confronted with difficulties, those individuals with a low confidence in their abilities are easily discouraged whereas more confident students will intensify their efforts (Igbaria & Iivari 1995). This concept was extended to the context of computer software. Studies in the nineties already showed the important role of computer-SEF in performances using information technologies (e.g. Compeau & Higgins 1995, Gist, Schwoerer & Rosen 1989). Individuals with a high computer SEF use the computer more, derive more of their use of computers and are able to exploit management support better (Igbaria & Iivari 1995). While computer SEF means general confidence in one's own abilities to work with a computer, independent from specific programmes or software, specific computer self-efficacy or programme-related self-efficacy describes one's own beliefs about being able to operate a specific software like for example a certain DGS (Agarwal, Sambamurthy & Stair 2000). It is yet unknown if persons with a higher programme-related SEF also benefit more of learning mathematical modelling with the use of a DGS than persons with a lower SEF concerning the development of modelling competencies. This could be the case because the former perhaps take more advantage of the tools a DGS offers while searching for mathematical models or adapting them. Additionally, user attitudes towards the computer or specific software can moderate the outcome of training programmes (Torkzadeh, Plfughoeft, & Hall 1999) or the individual's SEF (Torkzadeh & Dyke 2002). It is yet unknown, if the attitudes towards software also moderate the outcome on programmes where not the computer usage itself is trained, but the computer just serves as a medium to learn, e.g. mathematical content. Therefore, we take a closer look on the following questions in this paper:

RESEARCH QUESTIONS

1.) How do students perceive the DGS GeoGebra when learning mathematical modelling and especially, what difficulties do students encounter?

2.) Does the students' programme-related SEF or their attitudes towards the software change when learning mathematical modelling with a DGS?

3.) Is there a relationship between students' programme related SEF or their attitudes towards the used software and their growth of modelling competency when learning modelling with a DGS?

METHODOLOGICAL BACKGROUND

Design of the Study

To answer these research questions, we conducted an intervention study with a pre- and post-test as well as a four-lesson intervention in which students worked on geometric modelling tasks with the help of the DGS GeoGebra. A total of 328 grade 9 students in 15 different classes took part in this study, which was carried out in their regular mathematics lessons. During four consecutive math lessons, which were held in computer labs at their schools, the students worked in pairs on a geometric modelling task that was implemented in GeoGebra. Even though the participating teachers had to make sure that their students had already worked with the software GeoGebra before the beginning of the project, the teaching unit began with a short revision of useful symbols and constructions in GeoGebra to ensure basic knowledge about possible commands in the software. After this programme related revision, students worked independently on different modelling problems with the software. At the end of the respective lessons different solutions were projected and discussed. Before and after the teaching unit all students filled out a modelling test which measured their modelling competencies and a questionnaire concerning their confidence in their abilities to operate GeoGebra and their attitudes towards this software.

During the teaching unit, the students worked on four different modelling tasks. While the first of the used tasks was structured by different instructions and served as an introduction into the various steps of the modelling process, the remaining three tasks were rather open problems with several correct answers. For example, one of the tasks dealt with finding market areas of supermarkets in Berlin to determine where a new branch could be opened. With the help of GeoGebra, it is possible to try out different models. For example students may neglected the network of roads and find areas by constructing midperpendiculars between different branches of the supermarket. Alternatively, the existing roads can be taken into account and assumptions on the number of residents in different streets can be made. In this case it is sensible to choose polygons as market areas which can be adapted to their assumed number of residents.

All lessons started with a short presentation of the problem by the teacher to the class. Following, students worked in pairs on one computer. They had both a working sheet which presented the task in a written format and a DGS-file that contained the necessary graphics, e.g. a map of Berlin in the task described above. Students were asked to use the DGS GeoGebra and to write down their solution processes as comprehensible as possible. After a working time of approximately 25 minutes, several students presented their results or suggestions with the help of a projector. The whole class discussed and compared different solutions with a special focus on the different steps in the modelling process.

During the intervention, the teachers gave as little help as possible but gave students freedom to work independently and to make their own decisions. If the teachers intervened, they mostly gave strategic help or helped with issues with the software, e.g. helping to save the files in the right places. To prepare teachers for the intervention they were instructed with detailed materials including lesson plans in several meetings. During these meetings they were also prepared to typical questions and sensitized for students' modelling processes.

Questionnaires and Test Instrument

To assess students' programme-related SEF, which is their confidence in their own tool competency, an adaption of the CUSE-D questionnaire (Spannnagel & Bescherer 2009) was used. Since this questionnaire, originally developed by Cassidy and Eachus (2002), aims to measure general computer-related SEF, but not the confidence in one's own ability to operate a specific software, the used items were adapted to be more specifically related to GeoGebra. Students had to express their agreement to ten statements like "I think working with GeoGebra is easy" or "I think of myself as a skilled user of GeoGebra" on a rating scale with six categories. The lowest score of 10 points implied no confidence in the own tool competencies, the highest score possible of 60 points implied a very high confidence. The internal consistency of this scale was Cronbach's α=0.82 for the pre-test and α=0.92 for the post-test. The attitudes towards learning with GeoGebra were measured analogously. Exemplary items are "GeoGebra is a good help for learning" or "Using GeoGebra makes learning more interesting". Even though only five items were used, the reliability for this scale was as good as for the first with Cronbach's $\alpha = 0.87$ for the pre-test and $\alpha = 0.90$ for the post-test. The minimal score of 5 indicates strong disapproval of the software whereas the highest score of 30 indicates strong approval and a very positive attitude towards the software.

The modelling competencies were measured by a newly constructed test instrument that consisted of multiple choice or short-answer questions and focused on different steps of the modelling process. To avoid that students had to answer to the same modelling items twice but nevertheless to be able to use the same items in the pre- and in the post-test, a multi-matrix design was used, which had to be evaluated within the frame of Item Response Theory. With the help of this theory, Weighted Likelihood Estimators for the modelling sub-competencies at the different points of measurement could be estimated. As explained above, the steps of the modelling process differ regarding the use of a DGS. Therefore, we analysed two different aspects of modelling competency separately: On the one hand the sub-competency Mathematising/Validating (MV) in which the DGS was used to build, try out and compare different mathematical models and on the other hand a sub-competency Simplifying/Interpreting (SI) where the DGS did not play an equally active role. The estimators for these two dimensions could be determined with a reliability of $\alpha = 0.70$ for SI both in the pre-and post-test, and $\alpha = 0.71$ for MV in the pre-test and $\alpha = 0.73$ in the post-test. 40 % of the tests were rated by two independent coders. The interrater-reliability lay within a range of $.81 \leq \kappa \leq .95$ (Cohen's Kappa). Both the modelling test and the questionnaire were answered within 45 minutes in the math lessons directly before and after the teaching unit. For these lessons, no computers were needed since both the test and the questionnaire were purely in a paper-pencil format.

Interviews

During the teaching unit, the desktops of student pairs from six different classes ($n = 12$) were filmed and their conversations were recorded. These recordings and films were analysed to identify scenes where the students encountered difficulties during the modelling process. After the intervention, six pairs of students were confronted with their respective scenes and questioned in a semi-structured interview. The focus of this interview was on difficulties the students perceived while working on modelling tasks with GeoGebra, on their strategies to overcome these difficulties as well as on their attitudes towards GeoGebra and digital tools in general.

Methods of Evaluation

To evaluate the interviews, they were transcribed and coded in accordance with the summarizing qualitative content analysis. To do so, all transcribed interviews were line-serially analysed to inductively build different categories (Deeken 2016). In a final analysis, all interviews were coded.

Since the intervention took place in the regular classes, the quantitative data is structured in clusters. Students who are in the same class are likely to be more similar to their classmates than to students from different classes. Ignoring this structure when statistically evaluating the intervention would lead to distorted standard errors and thus to incorrect tests of significance. That is why we decided to correct these errors by using the programme Mplus and the *type = complex*-Option. With Mplus, we calculated Wald's t-tests to analyse possible change in programme-related SEF and attitudes towards the programme from pre- to post-test. To analyse a possible relationship between the growth of modelling competency and SEF and attitudes towards the software respectively, we calculated multiple regressions using the post-test values as dependent variable and pre-test values, gender, SEF and attitudes toward GeoGebra in the post-test as covariates.

FINDINGS

Results of the Interviews

The analysis of students' remarks in the interviews concerning their perception of GeoGebra revealed several different categories: required working time, insecurity, calculating device, operation, precision and usefulness. Students saw a connection between the use of the DGS and the time they spent working on the task. While some students found working with the DGS to be "quick and easy", others said looking for complex models was "time-consuming and complicated". The latter was especially the case when students still felt insecure with the software. In those cases, students were for example "slightly annoyed by those appearing numbers you had to hide by clicking on the buttons on the right". Often, these difficulties were due to missing knowledge about the geometric rules of constructions that lay behind the software's commands. All students agreed that working with the software became easier once they felt more familiar with it. They also perceived the software supportive of calculations. Some students remarked that they "had no clue how to calculate" the surface of a non-regular polygon without the software. They found the DGS to be practical, some saw an even bigger potential for the software in more complex tasks than those used in the study. Other students stressed the software's precision. One student compared her work in GeoGebra with constructions on paper: "A pencil goes blunt and you have to sharpen it while working. And if you want to erase it, it does not go away completely. That's much easier with GeoGebra". The aspects of changing models, adapting models or restarting a modelling process are also remarked by several students. "You can delete things quickly without having to restart completely" a student says. He goes on: "We went through the different options in the programme to find something to model the figures in the best way possible". Like him, several students mentioned the opportunity to be inspired by the commands implemented in the software while searching for suitable mathematical models. They "just tried out different things without having to ask someone" and "were able to find solutions on [their] own". Thus, the mathematics lessons became "a welcome change to regular math classes" and GeoGebra "a sound assistance".

Change in programme-related Self-Efficacy and Attitudes

A total of 289 students answered to all items measuring the programme-related SEF in both the pre- and the post-test. The confidence in tool competencies increases from a mean of 38.63 (SD = 8.12) in the pre-test to a mean of 44.41 (SD = 9.71) in the post-test. The Wald's t-test reveals that this difference of 5.46 is significant ($t(1) = 74.64, p < .001$). The effect size Cohen's $d = 0.61$ indicates a medium effect. The attitudes towards the programme remain relatively stable with a mean in the pre-test of 20.26 (SD = 5.74) and 20.42 (SD = 6.42) in the post-test. The Wald-test shows that this difference is not significant ($t(1) = 0.182, p = .670$), Cohen's $d = 0.03$ also indicates no effect. It thus can be stated that the four-lesson intervention where students worked on modelling tasks with a DGS lead to a significant improvement in their programme-related SEF while their attitudes towards the software remained the same.

Table 1. Descriptive Statistics for Pre- and Post-Test

	pre-test			post-test		
	N	M	SD	N	M	SD
SEF	277	38.86	8.06	308	44.45	9.49
Att	282	20.62	5.65	311	20.66	6.28
MV	320	0.00	0.72	320	0.11	0.80
SI	320	-0.04	0.75	320	0.01	0.84

N =Number of participants; M = Mean; SD = Standard Deviation; MV=Mathematising/Validating; SI=Simplifying/Interpreting; SEF=programme-related self-efficacy (measured in post-test); Att = attitude towards the software (higher value=more positive attitude, measured in post-test);

Relationship between Modelling and programme-related Self-Efficacy or Attitudes

As it can be seen in Table 2, there is a significant correlation between the programme-related SEF and the competency Mathematising/Validating (MV), both measured after the intervention, but not between the programme-related SEF and the competency Simplifying/Interpreting (SI). The more confident a person in their abilities to operate the software is, the better their result in MV in the post-test is. But this is also valid for their results in the pre-test, as the correlation between MV in the pre-test and SEF is significant as well. Persons who feel more confident in their tool competency after the unit also achieved a higher score in the pre-test. Concerning SI the correlation with SEF is significant for the pre-test only. Apparently, students who have a higher competence in SI at the beginning of the teaching unit also have a higher programme-related SEF. This seems to change during the teaching unit so that at the end no relationship between SEF and SI-competencies can be seen.

The attitudes towards the software and the programme-related SEF are strongly correlated. The more confident a person in their own competencies in using the software is, the more positively they see the software. It can also be seen that neither MV nor SI are correlated with gender but SEF and attitudes are. Boys tend to be more confident in their own abilities to operate the software and see the software more positive.

Table 2. Correlations of the used variables in the regression models

	MV _post	MV_pre	SEF	Att		SI_post	SI_pre	SEF	Att
MV_pre	0.41***				SI pre	0.46***			
SEF	0.20***	0.15**			SEF	0.10	0.15**		
Att	0.07	0.03	0.62***		Att	0.04	0.06	0.62***	
gender	0.12	0.05	0.26***	0.16**	gender	0.09	0.06	0.26***	0.16**

**p<0.01; ***p<0.001 (MV=Mathematising/Validating; SI=Simplifying/Interpreting; SEF=programme-related self-efficacy (measured in post-test); Att = attitude towards the software (higher value=more positive attitude, measured in post-test); gender: 1= boys, 0 = girls)

To analyse the influence of programme-related SEF and the attitudes towards the software on the modelling competencies independently from differences of competencies that already existed before the teaching unit began, two multiple regression-models were calculated. With help of the first model we analysed if programme-related SEF or attitudes towards the software were significant predictors of the achievement in the post-test in the dimension MV when controlled for both achievement in the pre-test and gender. The second model examined analogously their influence on the independent variable SI.

As can be seen in Table 3, only the score in the pre-test and the programme-related SEF are significant predictors of the achievement in the post-test concerning MV. Persons who feel more confident in using GeoGebra also improved their competencies in MV more, regardless of their gender. The attitude towards the software was no significant predictor of the post-test achievement in MV. The development of this modelling competency seems to be independent from the students' perception of the programme. Even when they did not recognize the software as a useful instrument for learning, they were able to build up the modelling competency MV by modelling with it. The standardized regression weight indicates a small effect size (β=.15). And on the other hand, a positive view on the software did not automatically lead to a stronger improvement in the modelling competency. This is also valid for the competency SI.

For the achievement in SI in the post-test, only the score in the pre-test is a significant predictor. Persons who are more confident in their tool competencies thus do not achieve higher scores in the SI part of the modelling post-test, when adjusted for the pre-test scores. Equally the attitude towards the software had no influence on their achievement either. With these regression models 19.4 % and 22 % respectively of the total variance can be explained.

Table 3. Multiple Regressions on MV_post and SI_post

model	coefficient	b	SE	β	p	R^2
1 (criterium: MV_post)	Intercept	-0.37	0.23	-.46	.12	19.4 %
	MV_pre	0.43	0.08	.39	< .001	
	SEF	0.01	0.00	.15	<.001	
	Att	-0.01	0.01	-.05	.54	

	gender (1=boys)	0.12	0.10	.15	.23	
2 (criterium: SI_post)	Intercept	-0.10	0.17	-.12	.57	22.0 %
	SI_pre	0.50	0.05	.46	<.001	
	SEF	0.00	0.01	.03	.64	
	Att	-0.00	0.01	-.02	.72	
	gender (1=boys)	0.11	0.08	.13	.18	

MV = Mathematising/Validating; SI = Simplifying/Interpreting; SEF = programme-related self-efficacy (measured in post-test); Att = attitude towards the software (higher value = more positive attitude, measured in post-test); gender: 1= boys, 0 = girls

Summary, Discussion and Outlook

To sum up the findings it can be stated that students did recognize possible benefits of working with a DGS as they stressed the software's precision und usefulness. But they also experienced difficulties, mainly due to missing either mathematical or software-related knowledge. Students stated that after having worked with the software during the four intervention lessons they felt more secure and more confident in their tool abilities. This impression can be confirmed by the quantitative data. We have seen that even the short period of four lessons in which students worked with a DGS led to a significant improvement in their programme-related SEF that was sustained even three month after the teaching unit. The attitudes towards learning with the software though remain stable throughout all points of intervention. Apparently students did not see the software more positively even though they felt more secure with it.

Comparing the two dimensions of modelling competency, only the development of the competency MV is influenced by the programme-related SEF when modelling is learned with the help of a DGS. This is in accord with the theoretical considerations that there are different phases of the modelling process where the software can play different roles. While trying out different models, adapting them or searching for alternative useful mathematics the DGS plays an important role. It serves not only as tool to visualise models, but it also gives inspirations on what kind of mathematics could be worth trying out to find a solution to the given problem. A possible explanation for the relationship between programme-related SEF and the development of MV is that persons who feel confident in the software can concentrate more on the step of mathematising or validating. Perhaps those persons can profit more of the benefits that the software offers which then leads to a greater improvement in the modelling competence. This assumption is supported by the results of the qualitative study where students recognised possible benefits of the software but also saw the need of basic knowledge in operating the DGS. Nevertheless, students' difficulties were not solely to be attributed to the software. The combination of the qualitative with the quantitative data showed that especially those students who improved their competencies in mathematising and validating often also named difficulties that resulted from the task itself and not from problems in operating the software. But of course the students who were interviewed is just a small sub-sample and certainly not representative. The intervention study, even though conducted with a large number of students was limited to students of higher-achieving schools in grade nine. This was mainly due to practical reasons as this group of students was most likely to have already worked with a DGS. A focus on complete novices or experts in the software when modelling with a DGS might be useful to reflect the results found in this study.

But nevertheless, this study gives a first insight into the complex interplay between modelling and factors like programme-related SEF and attitudes when modelling with a DGS. Yet there is still a need for research on how modelling with digital tools can be successfully learned, which premises should be fulfilled so that digital tools can be used in a profitable way when learning modelling and finally on the effects of digital tools on the development of modelling competencies.

An important point also lays in the design of tasks when digital tools are available. In our research, the used tasks could still be solved without the use of a digital tool. This reflects the usual practice in classrooms of just expanding students' tools in solving tasks and is the best basis to understand what changes in a working process are caused by the software. Equally interesting is the question how modelling processes change, when the problem cannot be tackled without a digital tool. Perhaps in those tasks the role of computer competencies has to be taken in account even more.

In our up-coming studies, a special focus will be given to the role of difficulties when modelling with a DGS. As the interviews in this study have shown, students remark problems that could be traced back either to a lack of software-knowledge or to barriers in the modelling process independently from the tool. As the confidence in tool-competencies seemed to have a bigger impact on the development of the competencies mathematising and validating than on other competencies, the hypothesis arises that in those phases of the modelling process more software-related difficulties might occur. Detailed observations of students modelling processes with special focus on their difficulties will give more insight into the different role of the software during different phases of modelling. The observation of students' modelling processes in this study already revealed interesting scenes where difficulties in using the software (e.g. how to construct a circle) led to a deeper mathematical understanding of the problem (e.g. is it generally possible to construct a circle in the given situation). We hope by this means to reach a better understanding of how a digital tool and the process of learning how to use a tool can be used in a promising way to foster modelling competencies.

REFERENCES

Agarwal, R., Sambamurthy, V., & Stair, R. M. (2000). Research Report: The Evolving Relationship Between General and Specific Computer Self-Efficacy—An Empirical Assessment. *Information Systems Research, 11*(4).418-430.

Bandura, A. (2012). *Self-efficacy: The exercise of control*. New York, NY: Freeman.

Blum W. (2011) Can Modelling Be Taught and Learnt? Some Answers from Empirical Research. In: Kaiser G., Blum W., Borromeo Ferri R., Stillman G. (eds), *Trends in Teaching and Learning of Mathematical Modelling. International Perspectives on the Teaching and Learning of Mathematical Modelling*, Vol 1. (p.15-30). Dordrecht: Springer.

Blum, W. (2015). Quality Teaching of Mathematical Modelling: What Do We Know, What Can We Do? In: Cho, S.J. (Ed.), *The Proceedings of the 12th International Congress on Mathematical Education - Intellectual and Attitudinal Challenges* (p. 73-96). New York: Springer.

Borromeo Ferri, R. (2006). Theoretical and empirical differentiations of phases in the modelling process. In: *Zentralblatt für Didaktik der Mathematik*, 38 (2), 86-95.

Cassidy, S & Eachus, P. (2002). Developing the computer user self-efficacy (CUSE) scale: Investigating the relationship between computer self-efficacy, gender and experience with computers. *Journal of Educational Computing Research*, 26(2), 169-189.

Compeau, D. R., Higgins, C. A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quart*. 19. 189–211.

Deeken, J. (2016). Modellierungsprozesse mit GeoGebra aus Sicht von Schülerinnen und Schülern in Klasse 9 – eine qualitative Interviewstudie. Unpublished Masterthesis. Westfälische Wilhelms-Universität: Münster.

Gist, M., Schwoerer, C. & Rosen, B. (1989). Effects of alternative training methods on self-efficacy and performance in computer software training. *Journal of Applied Psychology*, 74(6). 884–891.

Greefrath G, Siller H, Weitendorf J. (2011). Modelling considering the Influence of Technology. In Kaiser G, Blum W, Borromeo Ferri R, Stillman G (Eds.), *Trends in Teaching and Learning of Mathematical Modelling* (p. 315-329). Dordrecht: Springer.

Igbaria, M., & Iivari, J. (1995). The Effects of Self-Efficacy on Computer Usage. *Omega*, 23 (6), 587-605.

Siller, H.-St. & Greefrath, G. (2010): Mathematical modelling in class regarding to technology. In Durand-Guerrier, V.; Soury-Lavergne, S. & Arzarello, F. (eds.), *Proceedings of the Sixth Congress of the European Society for Research in Mathematics Education* (p. 2136-2145). Lyon: IRNP.

Spannagel, C. & Bescherer, C. (2009). Computerbezogene Selbstwirksamkeitserwartung in Lehrveranstaltungen mit Computernutzung. *Notes on Educational Informatics - Section A: Concepts and Techniques*, *5*(1), 23-43.

Torkzadeh, G., & Van Dyke, T. (2002). Effects of training on Internet self-efficacy and computer user attitudes. In: *Computers in Human Behavior*, 18(5), 479–494.

Torkzadeh, G., Plfughoeft, K., & Hall, L. (1999). Computer self-efficacy, training effectiveness and user attitudes: an empirical study. *Behaviour & Information Technology*, 18, 4 299–309.

Sedig, K. & Sumner, M. (2006). Characterizing Interaction with Visual Mathematical Representations. International Journal of Computers for Mathematical Learning, 11, 1–55.

Mackrell, K. (2011). Design decisions in interactive geometry software. ZDM Mathematics Education, 43, 373–387.