

New Challenges in Developing Dynamic Software for Teaching Mathematics

Zsolt Lavicza and Balazs Koren

Report

The principal aims of the discussion group were to discuss the development of a variety of mathematical software applications developed during the past decades. Among the most often utilised software types in education are Computer Algebra Systems (e.g. Derive, Mathematica, Maple, Maxima), Dynamic Geometry Systems (e.g. Cabri Geometry, Geometer's Sketchpad, Cinderella, GeoNEXT, GeoGebra), Spreadsheet and Statistics Software (e.g. Excel, SPSS, Fathom, R). Most of this software has been designed by keeping in sight primarily their usability for research purposes while others were predominantly aimed for their use in teaching. In the recent years we could observe, among others, three important trends in the development of these software tools: (1) Designers of research oriented software products started to involve features and support for educational purposes; at the same time teaching oriented software have been becoming increasingly more powerful so their use in some research is increasing; (2) The distinction between different types of software has begun to blur as many products integrate features from other types of software; (3) The computer platforms are diversifying; with the appearance of smart phones, tablets, and Interactive Whiteboards (IWB) in recent years, as well as online services such as Wolfram Alpha, challenging the design and development of mathematics software.

Organizers Co-chairs: Zsolt Lavicza (UK), Markus Hohenwarter (Austria); Andrian Oldknow (UK), Tolga Kabaca (Turkey), Kyeong Choi (Korea); Liasion IPC Member: Frederick Leung (Hong Kong).

Z. Lavicza (✉)
University of Cambridge, Cambridge, UK
e-mail: zl221@cam.ac.uk

B. Koren
Eotvos Lorand University, Budapest, Hungary
e-mail: balazs.koren@gmail.com

The Discussion Group aimed to elaborate some of the outlined issues with a mix of short presentations, questions, and reflections from the audience. DG-10 was divided into two sessions both of them were attended by approximately 60–70 participants.

Session I

In the First session Zsolt Lavicza and Balazs Koren outlined the aims of the session and set a schedule for presentations and discussions. The first presentation was given by Balazs Koren, Hungary introducing the different software available and used in mathematics teaching and research. Balazs wanted to have this presentation to challenge thinking of the audience and cluster software into tentative groups that was supposed to be dissected or rearranged into new categories and groups during the sessions. In the past decades, three kinds of tools were mainly used in schools: desktop software, specialised handheld calculators and more recently tablet and mobile phone apps, as well as web-based applications assisting mathematics teaching and learning. However, this grouping is getting outdated and the community should develop new clusters and characterizations to advance software applications and related theories. It seems that the borders between categories are getting more and more overlapped and we are converging towards more complex and adaptive systems in the near future.

The second presentation was given by Tolga Kabaca from Turkey. Dr Kabaca described his experiences with the mathematics community using a variety of software in Turkey. He emphasized that it is necessary to allow teachers to develop their own teaching applications, but at the same time there should be a structured system as well as training that allow them to bring technology into the classroom. In addition, Dr. Kabaca mentioned the importance of getting feedback for both software and material developers directly from the software tools.

Peter Boon from the Freudenthal Institute, Netherlands described the needs for extended environments around mathematical software. It is getting common to embed mathematical software into Learning Management Systems (LMS) or to a so called Digital Mathematical Environment (DME), which are enabling teachers and students manage learning within and outside classrooms, offer assignments of problems and collect data from their solutions. Developing such LMS and DME systems is difficult and may take years of improvements and modifications as well as needs an interaction with the used mathematics software environments.

Chris Sangwin, UK outlined his experiences in developing assessment tools for mathematics. STACK is an assessment environment developed by Dr. Sangwin including a number of own solutions, but at the same time drawing on the resources of other mathematical software. Assessment is one of the most controversial and difficult issues in today's educational debate so that creating an environment is challenging and risky. However, such environments are necessary for the educational system and ample thoughts have been invested into such developments.

The discussions during the sessions reflected the topics of the presentations. At the end there were suggestions for questions to be further investigated through debates and research:

- How do we or should we classify software applications?
- What does Dynamic means in a software environments?
- Do the use technology to enable mathematics learning, if yes, how?
- Do software need to offer and restructure social dynamics in classrooms and on the web?
- We need to emphasize the pedagogical uses of the software and develop them accordingly to enhance further opportunities for learning.

Session II

We consider this session as a historical event as all widely used software creators or leaders of their teams were represented in the room. Jean-Marie Laborde (Cabri), Ulli Kortenkamp (Cindarella), Zsolt Lavicza and Balazs Koren (GeoGebra), and Nicholas Jackiw (Geometer's Sketchpad).

The session also started with the presentation of Ulli Kortenkamp, Germany, who highlighted the difficulties and processes in the development of mathematical software. Dr Kortenkamp emphasized that there are a number of issues could arise when mathematical theories needed to be implemented in a computer environment. For example, matching Euclidian and Hyperbolic geometry into a single software could be challenging thus the community of mathematicians and software developers need to have forums to discuss possibilities for implementation.

Tatsuyoshi Hamada, Japan talked about a wide range of software developed in Japan and the difficulties of their spread across groups and universities. Professor Hamada created a downloadable live Linux application called Math Libre, which is a collection of freely available software tools for mathematics teaching and research. Trough this collection the authors aims that schools, teachers, and students can choose the best applications fitting their needs in education. The project contributed to the involvement of technologies in the curricula in many schools around the world.

Jean-Marie Laborde, France stressed the importance of quality and the mathematical correctness of software development. Professor Laborde described that software development is a costly process and needs to be done in a complex way to ensure the correct mathematical background of the underlining processes within the software. Thus, it is important that while choosing a software must be made based on quality and rather than economy.

Finally Zsolt Lavicza, UK outlined the development of an open source project and the importance of a community surrounding the software. GeoGebra has become a successful mathematical tool, because teachers and students found them on the Internet and started to contribute to both software and material development.

Due to the large user base and the responsiveness of developers to the requests of the users the software was developing quickly and attempt to correct the problems arising during its use.

The debate after the presentations initiated further ideas and questions:

- How we can deal with infrastructure issues in schools, in particular in lesser-developed regions in the world?
- How can we encourage education to use current rather than outdated technologies?
- Do we need to develop specialised or general software? Do we need to connect development with other fields such as with video game?
- How can we learn from the success of long existing and sustained software packages such as R?
- How can we best support LMS and DMS with mathematical software development?
- How can we deal with the complexity of mathematical software development?
- How can we set some guidelines for assessment with computers?
- Possibly we need flexible and customisable tools in the near future
- We need to produce more books and learning materials with different tools

DG-10 offered an inspiring environment to discuss issues for both developers and users of products. The presentations and reflections were fruitful, but because software use in mathematics education is still around the start line with the exception of some larger projects the session ended with more questions than answers. However, the beginning of such discussion is valuable and offered food for thoughts for participants and we believe already impacted the development of software.

Open Access This chapter is distributed under the terms of the Creative Commons Attribution Noncommercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.