Convergence on Mathematics Assistants

A new special track at

CADGME'09

July 2, 2009

Abstract

CADMGE attracts teachers and educators as users of Computer Algebra Systems (CAS) and Dynamic Geometry Systems (DGS) and of other mathematics assistants — which in turn attract developers of such systems.

This special track addresses developers and educators interested in future development of educational math systems; the issue is whether there could be some kind of "convergence" on the developments.

Many developments proceed at a high rate, nevertheless there is still educator dissatisfaction. This is in spite of the fact, that highly developed CAS and DGS are available for classroom use, that some systems begin to incorporate other concepts (DGS including CAS, spreadsheet, ...) and that other systems improve affinity to learning with support for activity-based inquiry approaches. Dissatisfaction might be concluded even from contributions to CADGME. For instance, one of them requests for a "CAD-dictionary" with suggestions for a "pedagogical CAS", another contribution advertises an "Algebra Drivers License" (are the systems too complicated ?).

Herein, please, find a survey on the papers within the special track, assembled under topics, which might be referred to in the discussions within the respective sessions. The questions noted for each session and each talk should not predetermine the discussions; they just try to relate the talks to each other.

Contents

1 On the convergence of geometry and algebra systems

Parallel session I, 11.Jul. 11:00 - 12:30
Session Chair: Pedrag Janicic,
`janicic@matf.bg.ac.rs`, University of Belgrade, Serbia and Montenegro

Speakers:

1. Walther Neuper, `neuper@ist.tugraz.at`, Graz University of Technology, Austria
2. Manfred J. Bauch, `manfred.bauch@obdimat.de`, Project ObDiMat, Germany
3. Steve Arnold, `steve@compasstech.com.au`, Compass Learning Technologies, Australia

Questions:
The three speakers all consider future development of DGS and CAS; the point of view are rather different: and it is up to discussion, whether some kind of convergence is in sight.

1. The special track "Convergence on Mathematics Assistants" assumes *common grounds for modeling mathematics* in logic-based software kernels. If such mathematics-engines reside in the background of various mathematics assistants, . . .
   (a) . . . is it desirable that the math-engine ensures/checks logical consistency between formulas on a worksheet (→Pt.2) ?
   (b) . . . might the stepwise construction of calculations as well as of geometric figures be one aspect for the integration of CAS and DGS (→Pt.3) ?
2. The title *"the future of the classical worksheet"* addresses a part of the user interface and its future – therefore two hypothetical questions:
   (a) If in 7 years *the traditional paper-based worksheet has not vanished*, rather has recaptured predominance more than tody, did something go wrong then ?
   (b) If in 7 years *electronic worksheet* has survived as the only version, are there some warnings to be foreseen ?
3. With respect to *"true" integration of the two environments* of DGS and CAS, . . .
   (a) . . . should this lead to *one* general purpose system or to an easy-to-download variety of special purpose systems ?
   (b) . . . how are issues of integrating the user-interfaces related to issues of conceptual integration ?

1. Walther Neuper: *Common grounds for modeling mathematics in educational software*. An Introduction to ConvMathAssist.
There is an abundant variety of mathematics assistants (MMs), successfully used in education. The MMs reflect the respective representations of mathematics objects in the various domains: geometry, algebra, numerical analysis and simulation, graph theory etc. And the MMs reflect various aspects and views of education.

Since, in principle, there is only one mathematics (with unified foundations commonly accepted today), and since all the variety of MMs should reflect that foundations, this talk asks the questions: (1) What are the common grounds for existing MMs ? (2) What are the principles the development of MMs might converge to ?

(1) Common grounds: Here a "step" is suggested as the minimal unit on common grounds; a step operates on an object (i.e. on an algebraic object like a term, an equation, a function, or on a geometric object, or on a graph like a dag, etc) within a logic and a

context. The step is related to at least one theorem and results in a transformed object and in an updated context. The talk will discuss how a step relates to rigorous foundations in logics, as well as to MMs in algebra and geometry (omitting graph theory et.al).

(2) Some principles for convergence: Since MMs represent formal mathematics, MMs might explicitly implement principles of computer mathematics. Since MMs "are (models of) mathematics", MMs might implement math knowledge in a human readable format (e.g. Coq, Isabelle, Mizar). Since formal logic is the basis of MMs but hard to learn, MMs might filter off for naive users. Since learners want to proceed on their own pace, MMs might uncover logical details on demand – providing continuous support from the introduction of variables up to academics.

The author looks forward to relate these suggestions to the variety of opinions presented in the working group "Convergence on Mathematics Assistants".

2. Manfred J. Bauch: Thoughts On *The Future of the Classical Worksheet*

A look at German classrooms shows that *the traditional paper-based worksheet has not vanished*. Quite the contrary, its deployment has increased. On the other hand, the notion of a worksheet also appears in the context of dynamic geometry and computer algebra systems.

Between these two extremes, one can find a large variety of modifications that give rise to the question about the future of worksheets: Will electronic and *paper-based version* coexist, will they merge or will *one form or the other vanish*?

We present several examples and discuss *elementary characteristics of a "worksheet"*, in particular their dependence on certain media.

3. Steve Arnold: Dynamic Integration of CAS and DGS: what does this mean ?

While we are seeing a growing range of software options which offer both computer algebra systems (CAS) and interactive or dynamic geometry systems (DGS) within the one envelope, in most cases these appear not to offer a *"true" integration of these two environments*: rather, the systems exist alongside each other with some nominal interaction. This session considers some examples of integration currently available, including GeoGebra, TI-Nspire CAS, Geometry Expressions and the new MathRider. It explores applications of such integration to the teaching and learning of mathematics, particularly at high school level, and finally poses questions regarding the possibilities for future development towards true dynamic integration between CAS and DGS.

2 Dynamic geometrie meets computer theorem proving

Parallel session II, 11.Jul. 14:00 - 15:30
Session Chair: Walther Neuper,
`neuper@ist.tugraz.at`, Graz University of Technology, Austria.

Speakers:

1. Markus Hohenwarter, `mhohen@gmail.com`, Florida State University, United States
2. Pedrag Janicic, `janicic@matf.bg.ac.rs`, University of Belgrade, Serbia and Montenegro
3. Makarius Wenzel, `makarius@sketis.net`, Technische Universität München, Germany

Questions:
This session attempts to relate DGS, which already contain components of symbolic computation, with Computer Theorem Proving (CTP) aiming at *"human-readable formal proofs"*.

1. In *the future the dynamic mathematics software* GeoGebra will expand to *symbolic algebra*.
   (a) Might the *symbolic algebra* be generalized to logic-based systems ? (→Pt.3a)
   (b) Are there plans to guide a user (→Pt.2c) through the steps of a construction ?
   (c) Can GeoGebra's implementation (in Java) be interfaced with the JEdit/Scala-trials (→Pt.3c) ?
2. Referring to the comparison of the *two families of theorem provers*:
   (a) How relate GCLC's language describing a construction and a *readable geometry proof* ?
   (b) How relate a *readable geometry proof* and a geometric construction (e.g. for the ortho-center of a triagle) ?
   (c) Can GCLC's language describing a construction be used to guide the users interaction (→Pt.1b) when trying this specific construction ?
3. Can the *general principles of organizing formal reasoning* be applied to geometric constructions (which might be considered "concrete proofs") ? In particular:
   (a) Can the *derived Isar language elements* include *"commands"* for geometric constructions ?
   (b) Is the *formal document model* general enough to also model geometric constructions ?
   (c) Might the JEdit/Scala-trials be interfaced with GeoGebra's implementation in Java ?

1. Markus Hohenwarter: GeoGebra - Past, Present, and Future of Dynamic Mathematics Software
The free dynamic mathematics software GeoGebra was originally based on ideas from interactive geometry and algebra systems and aimed to dynamically link them in one easy-to-use software package. GeoGebra's latest version now also includes a dynamic spreadsheet component and allows dealing with basic statistics concepts. Future versions plan to include *symbolic algebra* and 3D graphics extensions as well. These developments are driven by volunteers from all around the world, trying to give students and educators easy

access to dynamic mathematics software. In this presentation, I will discuss the emergence of GeoGebra and its open-source user and developer community, provide examples for applications, and dare a glimpse into the *future of such dynamic mathematics software*. With this presentation, I also hope to nurture future cooperation with other open-source initiatives interested in aspects of converging mathematics assistants.

2. Pedrag Janicic: Automated Geometry Theorem Proving: Readability vs. Efficiency

There are several very efficient methods for automated geometry theorem proving, including Buchberger's and Wu's methods. However, these methods are primarily algebraic methods and do not produce traditional and *readable geometry proofs*. On the other hand, there are some proving methods (e.g., based on coherent logic) that can prove geometry theorems in traditional manner, but the power of such provers is much less. Typically, these *two families of theorem provers* have different scopes and languages. Similarities and differences between these two families will be discussed and illustrated by the geometry tool GCLC and its built-in theorem provers.

3. Makarius Wenzel: The Isabelle/Isar framework as a "logical operating system"

Isabelle was originally conceived as a "logical framework" by L. C. Paulson in 1989. The meta-logic of Isabelle/Pure implements a bare minimum of higher-order natural deduction that enables users to specify a variety of object-logics, by declaring connectives and rules. This facilitates experimentation with basic logical calculi, although a realistic working environment will demand development of specific theory libraries (with numerous definitions and proofs, and add-on tools implemented in SML).

On top of the Pure framework, the Isar layer was added around 1999 by the author, in order to enable *"human-readable formal proofs"* (similar to Mizar). Implementing structured proof processing is a very demanding task, but our systematic approach allowed to single out *general principles of organizing formal reasoning* beyond the primitive calculi. The main Isar concepts are that of a theory and proof context, with generic data declared by user code, where the framework manages correctness wrt. monotonic extension, and transfer from one context to another.

This basic Isar infrastructure has been refined and extended further in recent years. There is now a general framework for "local theory specifications", which supports a variety of module concepts on the one hand (e.g. Isabelle locales, type classes, class instantiation, theory interpretation), and combines them with derived specification mechanisms on the other hand (e.g. inductive predicates, recursive functions).

The user can implement *derived Isar language elements* specifically as "proof methods" or "theorem attributes", or as arbitrary *"commands"* of the toplevel language. Ultimately, these language elements perform transactions on a *formal document model*, which are managed by the system in a valued-oriented fashion. The recent addition of implicit parallelism (with automatic proof scheduling) illustrates particularly well how the initial "logical framework" is evolving into a *"logical operating system"*. Thus it may serve as implementation platform for advanced mathematical assistants, beyond traditional interactive theorem proving.

3 Provers at the interface between university and school

Parallel session III, 11.Jul. 17:00 - 18:00
Session Chair: Walther Neuper,
`neuper@ist.tugraz.at`, Graz University of Technology, Austria.

Speakers:

1. Wolfgang Schreiner `wolfgang.schreiner@risc.uni-linz.ac.at`, Research Institute for Symbolic Computation (RISC), Austria
2. Cezary Kaliszyk

Questions:
The talks in this session present simplified access to computer theorem provers (CTP) and also simplified use of CTP by presentations of *problems in a way that is customary in undergraduate logic courses* to *quickly produce [. . .] formal proofs*. With respect to these simplifications – could they already justify the headline for this session ? In particular,

1. can the use of the provers be simplified (e.g. by *automatic applications of logical inference rules, and automatic simplifications/decisions*) such that simple proof can be found automatically (thus giving demonstrations to learners) ?
2. which kind of help does the student get in trying to complete a proof ?
3. could there be support for high-school teachers willing to try to introduce provers to their students ?

1. Wolfgang Schreiner: On Proving Assistants in the Classroom (and Elsewhere)
We describe our motivation for and experience with the use of the interactive proving assistant "RISC ProofNavigator" in courses on "Formal Methods" for master students of computer science and mathematics. The tool can be used to *quickly produce (by a combination of manual user interactions, automatic applications of logical inference rules, and automatic simplifications/decisions provided by an external satisfiability solver) formal proofs* as they typically arise in the verification of computer programs. In the same way the software can be applied in other scenarios where the goal is (on the basis of the language of predicate logic) education in formal modeling and reasoning, which are a human key qualification in many modern professions.

2. Cezary Kaliszyk, Teaching logic using ProofWeb
We will present the system ProofWeb, that allows teaching logic using an interactive web interface to the proof assistant Coq. We start by showing a web interface to proof assistants.

We then present an extension for teaching, that makes the full power of Coq available to the students, but simultaneously presents the logic *problems in a way that is customary in undergraduate logic courses*. We describe the supported natural deduction proofs styles, central tracking of the progress of the students, a database of logic problems, that also holds the students solutions and a parser that indicates whether the students used only the allowed inference rules of the logic.

Finally we show the possibilities of the use of the interface in collaborative proof development. We will describe our future project MathWiki that aims to create a wiki for formalized mathematics.

4 Concepts to design the structure of integrated systems

Parallel session IV, 12.Jul. 8:30 - 10:00
Session Chair: Walther Neuper,
`neuper@ist.tugraz.at`, Graz University of Technology, Austria.

Speakers:

1. Andre Heck `A.J.P.Heck@uva.nl`, Ton Ellermeijer `A.L.Ellermeijer@uva.nl`, Universiteit van Amsterdam, Netherlands
2. Christian Hirsch `csmc-wmu@wmich.edu`, Western Michigan University, United States
3. Matija Lokar `Matija.Lokar@fmf.uni-lj.si`, UL FMF, Slovenia

Questions:
Each of the three software-projects presents successful interfacing of several software components, unified user-interfaces and a wide scope of applications.

All three together present an impressive variety; this gives cause to a gedankenexperiment: Let some funding agency invite the three speakers to come with the technicians and didactics experts from their respective teams they need; and let them request to develop a system which integrates the ideas of all three projects. What might we expect to be the prevailing challenges:

1. technicalities in interfacing the software components, the *Scientific and Technical Open Learning Environment* with *CPMP-Tools* and *STACK* etc ?
2. the design of a unified user-interface to ensure smothly shifting from one component to the other ?
3. the clarification of the structure of how to present and access the various services and how to exchange data between them ?
4. relating the requirement, that formulas modeling some problem reside in a logically consistent context, to the variety of components (thus clarifying the prerequisite for a logic-based mathematics-engine in the background) ?

1. Andre Heck, Ton Ellermeijer: Mathematics Assistants: Meeting the Needs of Secondary School Physics Education

Coach is an activity-based, open computer environment for learning and doing mathematics, science, and technology in an inquiry approach, developed in the last twenty-five years at the AMSTEL Institute of the University of Amsterdam. It offers a versatile set of integrated tools for data collection, data analysis, modeling and simulation, and for multimedia authoring of activities. We present the STOLE concept, which is an acronym for *Scientific and Technical Open Learning Environment*, underpinning the design and implementation of Coach. It is an example of how members from the physics education research community came to convergence on tools for doing investigative work and achieved integration of tools.

Special attention goes further to the mathematical requirements of the learning environment and to the computer support of various representations of one and the same phenomenon or scientific concept. We also discuss one of the most complicating factors in the implementation of an integrated learning environment for mathematics and science, namely that mathematical concepts are not always used the same in these fields. Differences between the use of variables, functions, and graphs in mathematics and physics are briefly discussed, and consequences for the design of a general-purpose learning environment are addressed.

We exemplify what has already been realized in the Coach environment with regard to mathematical assistance, how these tools have been applied in real education, and which extensions of mechanized mathematics assistance are in progress or still awaiting functional specification.

2. Christian Hirsch: Innovation in Design, Access to, and Use of Software Tools for High School Mathematics

Since 1992, the Core-Plus Mathematics Project, with funding from the National Science Foundation, has been engaged in research and development of curriculum materials that interpret and implement professional recommendations for curriculum, instruction, and assessment in high school mathematics. Due to concerns for access and equity, the 1st edition materials were based on a modest technology assumption — students had access to graphing calculators for investigations and problem solving both in and outside of school. In current work on the 2nd edition, the contextual and mathematical problems that the curriculum is organized around and the learning expectations for students are such that it was desirable to augment graphing calculator use with computer tools. To meet this challenge and maintain access and equity, the project systematically explored the development of Java-based software that evolved into *CPMP-Tools* — a suite of general purpose and custom software tools, integrated with development of the curriculum materials.

The software suite includes tools for algebra (including a CAS), geometry (including a DGS), data analysis, and discrete mathematics (especially tools for constructing, manipulating, and analyzing vertex-edge graphs). The tools and their functionality are organized by Course needs to focus on the intended mathematics and to reduce the steepness of the learning curve. The tools are connected and share similar menu screens and user interfaces, thereby promoting integrated use and learning transfer from one tool to another. The tools are built using Java WebStart, which permits safe, easy, reliable distribution of software and updates across the Internet and on different types of computers. As public license software, CPMP-Tools is free to be used and further developed by others.

This session provides an overview demonstration of CPMP-Tools, discussion of design features and development decisions, and examples of how the software is being used to support mathematical explorations and problem solving.

3. Matija Lokar: Reuse of teaching materials

When we are working with e-teaching materials we much too often find that the authors of such materials, meant for the use of teachers in the teaching process, do not use the opportunities offered by the new technologies. All too often the materials are a monolithic block (or at least their main part is), constructed the way an ordinary book or workbook would be. This demands that the teacher take them as a whole, precisely in the order they were written in. Is that really necessary? Do all teachers need the same form of resources, do they want to use them in the same order, and do they want their students to see the same examples, do the same exercises? Why not use the possibilities that new technologies offer and at the very least give teachers the chance to adapt the materials to their own and their students' needs.

In the project Active Maths (http://am.fmf.uni-lj.si) we were basically concerned with making resources that can be changed and combined. Important part of our resources were tests and quizes, developed using different computer algebra systems and dynamic geometry systems. We also used system *STACK*, developed by Chris Sangwin from University of Birmingham. This project was a practical manifestation of the knowledge that was gained over the years of making and using e-teaching materials, namely that teachers want materials that can easily be changed and reused for their own specific needs. Different methods of use will be shown; from using simple links to the existing materials and direct copying of resources, to ways of changing, combining and using new resources.

5 A variety of contributions from Austria

Parallel session V, 12.Jul. 11:00 - 12:30
Session Chair: Walther Neuper,
`neuper@ist.tugraz.at`, Graz University of Technology, Austria.

Speakers:

1. Philip J. Ramsden `p.ramsden@ic.ac.uk`, Imperial College London, United Kingdom, Reinhard Simonovits `Reinhard.Simonovits@uni-graz.at`, Handelsakademie Grazbachgasse, Austria, Bernd Thaller `Bernd.Thaller@uni-graz.at`, University Graz, Austria

2. Josef Böhm `nojo.boehm@pgv.at` ACDCA & Technical University, Vienna, Austria, Eno Tonisson `eno.tonisson@ut.ee`, University of Tartu, Estonia

3. Christian Gütl `cguetl@iicm.edu`, Graz University of Technology, Austria, Alexander Nussbaumer `alexander.nussbaumer@uni-graz.at`, University of Graz, Austria

Questions:
This session collects Austrian contributions, which demonstrate a considerable variety of ideas. Nevertheless, we try to interrelate them:

1. *M@th Desktop* — could the system take profit from a single-stepping system, →Pt.3 ?

2. The *'dictionary' for the most important activities supported by a CAS*, could it really *form a base [. . .] to create a new "pedagogical CAS"*, →Pt.3 ?

3. The single-stepping system, . . .

   (a) . . . could it's content be expanded with content from Math Desktop transferred into html-format ?

   (b) . . . could it contribute to the dictionary (→Pt.2) with research on the language of formulas in mathematics, science and engineering ?

1. Philip J. Ramsden, Reinhard Simonovits and Bernd Thaller: Design of M@th Desktop (MD), Considerations of software design and how to teach and learn with M@th Desktop

M@th Desktop is an e-learning software based on Mathematica for teaching and learning Mathematics. The user of MD are teachers and students at high schools, universities of applied science, undergraduate students of science and technology,.

Design concepts of the learning environment MD

The basic elements of MD are palettes and notebooks. The palettes contain the "mathematics" necessary to deal with problems of a certain topic e.g. optimization. The accompanying notebook shows demonstrations, examples, exercises and a Test Your Knowledge section.

In the talk the role of different teaching styles with MD is discussed. The blended learning concept will take center stage.

The palette-notebook design of MD allows for a relatively simple handling of the courseware. Only a little knowledge of Mathematica is required. This is especially useful when teaching in big classes.

Customizability of MD: Each user, teacher and student as well, can design and develop own palettes. Students like to do it before test.

The teacher has MD's tools to his/her disposal to edit the existing notebooks or write his/her own content, prepare practice sheets, tests, etc. In M@th Desktop there are units for polynomial functions, exp, log and trigonometric functions, data fitting, differentiation, integration, linear algebra and statistics. The units cover the standard curriculum of EU and US high schools.

EU Projects

M@th Desktop has been adopted in 3 Comenius projects for high schools as teaching software. During the LTM Comenius 2.1-project in 2005-2008, coordinated by the University Graz, Austria, two further modules of MD were developed.

LTM homepage: http://ltm.uni-graz.at/, MD Homepage: http://www.deltasoft.at

2. Josef Böhm and Eno Tonisson: Do we need a CAS-dictionary ?

Many CAS users - especially those who use a CAS as an educational tool - are facing exciting times. There are several CASs available and the situation may change quite rapidly. The new CASs could appear and existing ones could fade away. For example, DERIVE is off the market and there are a couple of competitors for taking the leadership as a classroom tool.

Teachers have the choice between many products and they might try various products before deciding what to recommend to the school authorities, the parents, the colleagues and last but not least to the students.

We wonder if something like a 'vocabulary' or *'dictionary' for the most important activities supported by a CAS* (simplification, solving of equations and inequalities, drawing graphs of the functions etc.) might support the teachers and students as well when changing from one system to the other. The commands are very similar but often tiny differences in syntax and output of the results can cause problems.

The possible product (vocabulary, dictionary, ...) could *form a base for people who would like to create a new "pedagogical CAS"*. Also, it could be useful for the researchers who compare different CASs.

The product should be web-based, user-friendly and open for supplements. In a sense the multilingual dictionaries could be taken as the models. In our case, the commands of certain CAS form the "language" (or "dialect"). On the other hand there are several special needs.

We try to show some possible technical solutions. However, our main goal is to discuss the reasonability of such a dictionary in the conference.

3. Christian Gütl, Alexander Nussbaumer: Enhanced Personalized Learning Support of Computer Algebra Systems

Computer mathematic tools such as computer algebra systems and computer theorem provers have supported mathematicians, scientists and engineers by automating tasks for decades. Computer tools for numeric and symbolic mathematics has also gained increasing interest in different educational settings. Application scenarios in secondary education focuses mainly on the support of solving physical and engineering problems and e-assessment scenarios. Exploring and applying mathematic rules and theorems is a new and promising application scenario which raises novel issues in the education community. Although such application scenarios may support modern learning and teaching approaches, existing tools have not sufficiently used so far the great potential of computer mathematic tools the could provide. Focusing on computer algebra systems, they should not only track, assess and support students' single steps in solving mathematical problems, they could also identify which mathematical rules and theorems have been applied as well as which of the concepts can be successfully solved and which are challenging for the students. This features may be used the basis to build fine-grained user profiles about students' skills and link them to the mathematic concept space which can be used for personalized learning and assessment activities. In this talk (paper), we will discuss our approach and first findings in this idea which is built on the $\mathcal{ISAC}$single-stepping system and the Knowledge Space Theory modeling user profiles.