Macros and Modules in Geometry

Gert Kadunz, Klagenfurt (Austria)

Abstract: The paper highlights the importance of "macros" or modules for teaching and learning Geometry using Dynamical Geometry Software (DGS). The role of modules is analyzed in terms of "writing" and "reading" Geometry. At first, modules are taken as tools for geometrical construction tasks and as tools to describe and analyze these constructions. For proofs, decomposing a given geometrical statement may be supported by using prototypical pictures representing theorems of geometry ("modules"). Reading theorems into geometry and constructing proofs is still a major achievement of the student - which may be reached by using macros and modules as a major heuristic strategy.

Kurzreferat: Dieser Beitrag möchte die Bedeutung von Modulen (Makros) beim Lernen von Geometrie am Beispiel von Software zur dynamischen Geometrie beleuchten. Dazu wird die Rolle von Modulen als Mittel zum „Schreiben" und zum „Lesen" von Geometrie vorgestellt. Module werden damit als Werkzeuge gesehen, mit denen geometrische Konstruktionen hergestellt und geometrische Konstruktionen untersucht werden können.

ZDM-Classifikation: C30, C70, G10, N80, U70

1 Characteristics of Dynamical Geometry Software

In research on mathematics education (Didactics of Mathematics), "Dynamical Geometry Software (DGS)" is used as a generic term to describe a certain type of software which is predominantly used for the construction and analysis of tasks and problems in elementary geometry. Since around twenty years, software developers, mathematicians and didacticians of mathematics came up with a series of these products like "Cabri-géomètre" from France, the German "Euklid" and "Geolog", the US-American "Geometry's Sketchpad" and "Thales" developed in Klagenfurt/Austria. These and other pieces of software share certain features which allow to put them all in one category. Literature from mathematics education research offers three characteristic features to make a certain software for elementary Euclidean geometry a "DGS":

- "dragmode" as dynamical modeling of traditional tools from Euclidean geometry,
- "macros" to condense a series of constructions steps into one software command,
- "locus of points" to show the path of one or more points when dragging another point (cf. Graumann et al. 1996, p. 197).

In this paper, we concentrate on the "macro"-feature of DGS and will only marginally use and comment on the dragmode and locus of point feature (for dragmode and locus of point see Arzarello and Jahn in this volume).

2 Definition and Role of Module

2.1 Modules and Macros in Informatics

Contrary to what we did in the introduction of this paper, "module" and "macro" are not used as synonyms in Informatics. If users and/or developers condense a sequence of commands which is often used into one unit, one command, they define a "macro". It will be labeled by a clear name (a signifier) and can be used by this throughout the whole consecutive work. Internally and hidden from the user, a "macro-expander" will substitute the signifier by the initial sequence of commands. In the literature, macros are also described as programs not defined by a sequence of program code, but defined by prototypic construction. In the 1980ies, this feature was discussed under the heading of "programming by example (PbE)" (see the basic monograph Cypher et al. 1993).

In contrast to this, the word "module" is used in informatics: When describing the development of software, a module will be a rather large part of a software system, if not a sub-system, which together with other modules will make up for the whole system. Exchange between modules will be controlled by specified interfaces. The individual module is logically and functionally closed and normally hides its inner structure ("information hiding") and can be tested independently. Larger software products (like for instance MS-Word) offer the definition of macros by the user - which can be understood as "modules" and may be programmed by the developer and defined by certain input specifications and an inherent control structure.

What will be described below as "DGS-macros" is something in between modules and macros if viewed from the point of view of informatics. On the one hand, DGS-macros are defined by prototypic constructions, on the other hand they need a clearly defined interface and normally hide their inner structure from a potential user (see part 4 of the paper). But what about modules and macros from a didactical point of view ?

2.2 Modules and Macros in Didactics of Mathematics

Research in cognitive psychology reports that the cognitive ability of experts is heavily determined by their use of structured units and/or patterns of knowledge ("chunks"). This idea is also used by didacticians of mathematics (like Dörfler 1991; Dubinsky 1988) who also suggest to describe learning of mathematics in terms of thinking in patterns, blocks, modules and chunks. In these modules, knowledge is condensed into a unit available to the learner as a whole. Using modules implies a reduction of complexity within problem solving. Even if the word "module" is rarely used in scientific mathematics, mathematicians and learners of mathematics often employ modules. To give some examples, we mention algorithms to solve (systems of) equations, lots of famous formulae (like the "binomials") and especially the lemmas and theorems which reduce an often lengthy argumentation to a "simple" and easily used statement.

For learning mathematics, knowledge about using modules seems to be more important than detailed knowledge about the inner structure of a module. A conceptual and operative understanding seems to precede the knowledge how to construct a certain module (Dörfler 1991, p. 73). Using modules within software can be supportive for the modularity of problem solving and learning and doing mathematics. Using "procedures" to organize and structure programming can be an indication of such a modularization. Pocket calculators (with or without computer assisted algebra systems CAS) offer special keys for built-in modules.

As a by-product, use and development of modules in mathematical software force the developer or learner to give a name to the modules - thus implying the necessity to "speak" about mathematics. Following F. Schweiger (1992) one could take “module” as one of the "fundamental ideas" within learning of mathematics. With the perspective of P. Bender on fundamental ideas, modules should have a certain openness ("Weite" in the sense of logical generality), broad applicability ("Fülle" in the sense of multiple applicability in different sub-domains) and common sense (groundedness in everyday practice; see Bender&Schreiber 1985 p. 199). Given the length of this paper, these ideas of Schweiger and Bender&Schreiber cannot be elaborated.

2.3 Module s and Macros in Dynamical Geometry Software

The above mentioned definition of macros as “programming by example” describes the way how users of DGS produce the macros: They basically set up a linear program without control structure (like loops and if-then commands). Input and output parameters have to be exactly determined, the construction steps are not normally accessible to the potential and actual user of such a macro. To the user, the macro function is a black box producing defined output from defined input. Well-known examples of macros are regular polygons constructed from (endpoints of) a given segment or geometrical transformations like reflections or translations. The inversion at a given (fundamental) circle for instance seems only available in present school geometry with the help of an appropriate macro (see also part 3.2 of this paper).

A comprehensive, well-equipped set of macros (like the one from Tim Lister 1989 for hyperbolic geometry with Cabri-géomètre) turns DGS into an excellent tool for hyperbolic geometry (see Sträßer 2001, pp. 321-323). The very same set of macros also shows lack of control structure and input parameters with DGS-macros: Each and every use of the Lister-macros implies the explicit identification of the same fundamental circle by a mouse click. DGS-macros (at present) do not realize the identity of the fundamental circle in each macro-use within this model of hyperbolic geometry (see Sträßer 2002).

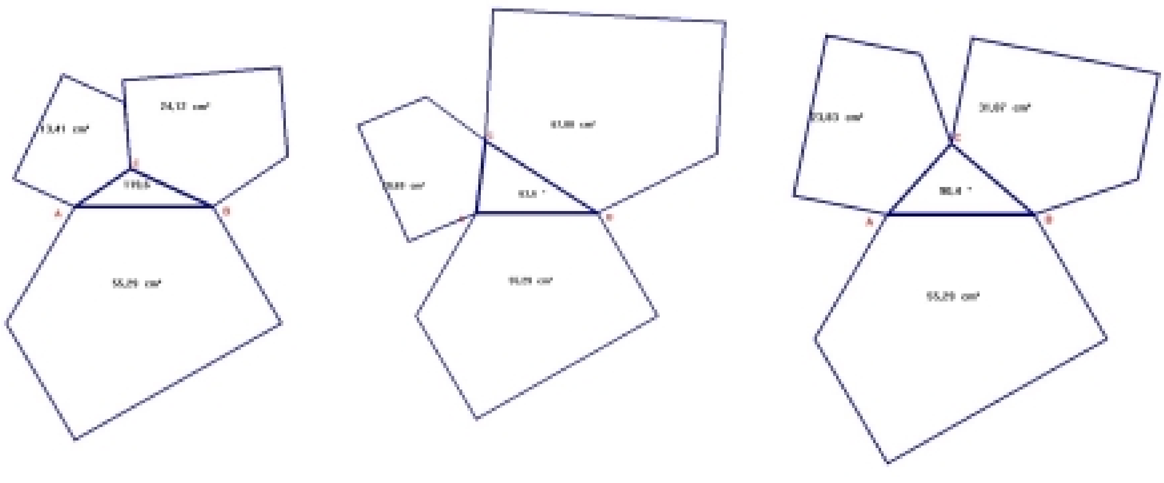
Looking back, it is clear that macros (or modules) are used for geometrical constructions. They save mouse-clicks and time in construction tasks by offering more complex construction tools than the traditional points, segments and circles. In order to get some deeper insight into the role of macros, the next chapter presents some examples to open up a perspective for some more theoretical insight and conclusions.

3 Examples

3.1 How to Generalize the Pythagoras' Theorem

One of the best known statements from school geometry, if not school mathematics, is the Pythagoras theorem on squares above the segments of a rectangular triangle. If one does not want to introduce it by mere stating the recipe, there is an opportunity for macros and the dragmode of DGS. After constructing squares above the segments of a non-constrained triangle, the learners may vary the points and observe the (areas of) the squares. Soon there will be configurations where the sum of two areas is obviously smaller than the area of the third square as well as there are positions where it largely exceeds the third area. Intuition may suggest that there are configurations with an identity of the sum to the third area. How can these configurations be easily characterized?

In order to construct this configuration a macro for squares above a segment (or its endpoints) may be very helpful and time saving if compared to a traditional construction of three squares. This macro can "naturally



Drawing 1

lead to a generalization of the Pythagoras theorem by macro-construction with different similar polygons above the segments of a rectangular triangle. For an exploration of this statement macros for (regular) polygons defined by one segment are nearly obligatory.

Drawing 1 shows the construction of such a (non-square) polygon together with three positions of the triangle when it is being dragged. It would be a nice learning experience to asks students to describe the conditions for equality of the sum of the area of two polygons to the third area (e.g. by means of an electronic worksheet).

3.2 Inversion of Conic Sections

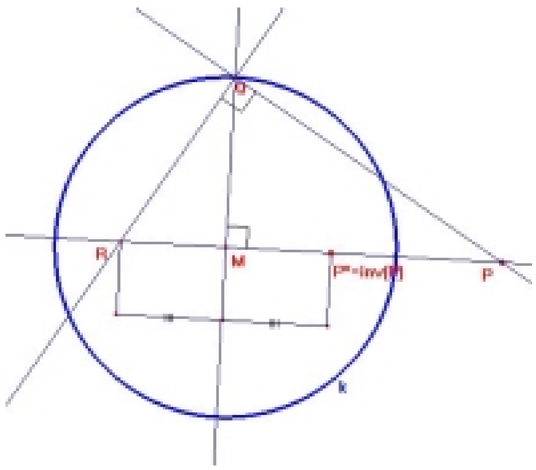
Similar to the generalization of Pythagoras, complex problems related to the inversion at a circle can only be explored using macros. The macro has to be defined in a way that it works homogeneously for points in whatever position (inside, on and outside the circle of inversion).

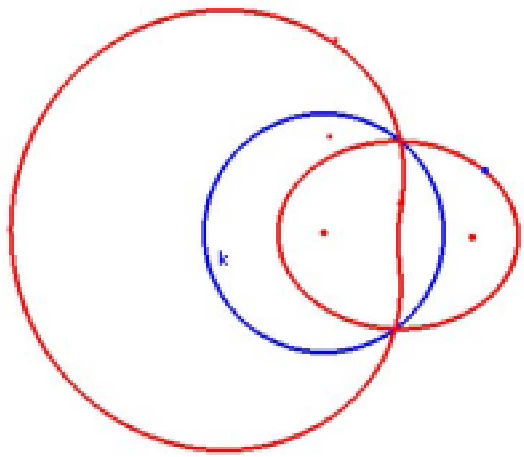
Drawing 2 shows the images of ellipses and hyperbola when being inverted at a given circle k - producing a whole family of curves not normally accessible and analyzed in school geometry but well-known in traditional geometry (for details see Kastner 2001).

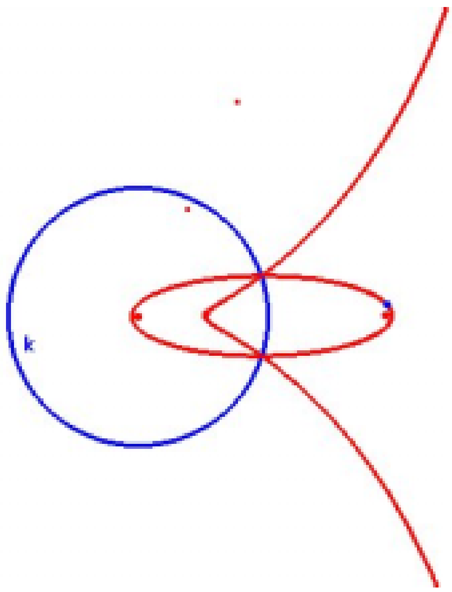
3.3 What do We Learn from the Examples?

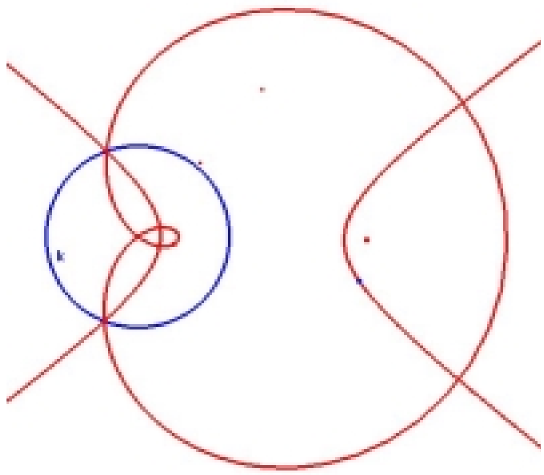
Both examples in part 3 necessitate macros for constructing the drawings in reasonable time and make the drawing available for inspection and exploration. In addition to this time economy, the macros allow to see the constructions in a new light: The use of the macro (and the dragmode) enables the learner to deeply analyze the dependency of the conic section and its transformation under the inversion. The use of a macro opens the access to a dynamic exploration and a new perspective on the Pythagoras theorem. The dynamic drawing offers the opportunity to concentrate on the respective magnitudes of (sums of) areas.

In all, the entire construction process changes with a new relation between the individual units of construction. It is not the point-wise relation of a point of the conic section and its image under inversion which comes to mind, but the entire image is seen in relation to its









Drawing 2

argument. So it is easier to relate characteristics of image and argument to one another - as for instance the two parts of the hyperbola going to infinity which imply a specific behavior of the image near the center of the circle of inversion. With the ellipse as an argument not going to infinity its image keeps off the center of the inversion-circle. It is only when the center of the circle is a point of the ellipse that the image of it goes to infinity. The quantitative/economy aspect here turns into quality of analysis.

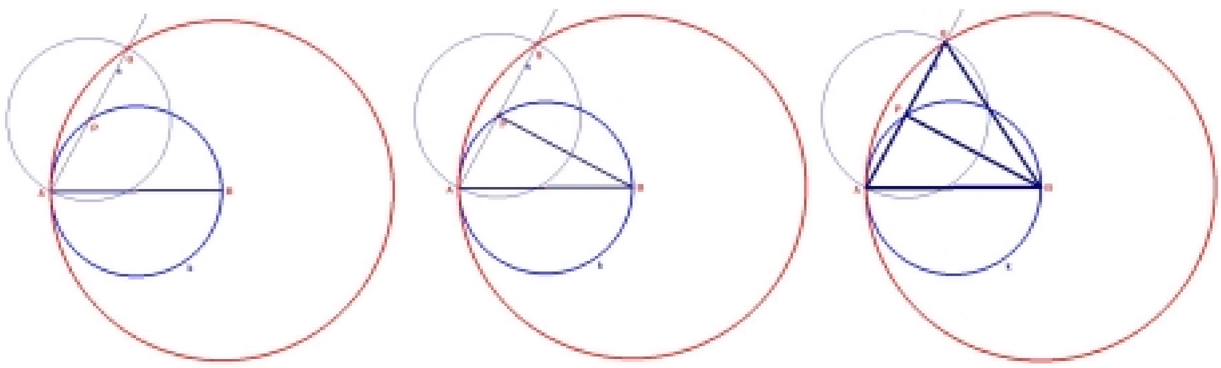
The combined use of dragmode and macros shows an additional aspect of macro-constructions: With the help of the dragmode, a drawing is changed while respecting and leaving unchanged the geometrical relations implied by the construction. Internally, for the software, this is only possible with an algebraic representation of the construction parallel to the geometrical - in DGS: graphical - representation. Input from the learner first changes the internal, algebraic representation which then controls the graphical representation. When varying a construction via dragging, the construction reacts as if the user would transpose an algebraic formula. In elementary algebra, it is not unusual to transform (only) parts of the algebraic expression and to group, to chunk together parts of expression to generate an algorithmic solution. In a construction task, macros have a similar function: they group a number of construction steps into one command and set up the whole construction using "geometrical expressions".

It is exactly here that the analogy comes to an end. In algebra, the transformation of given expressions hopefully comes to a stop in a solution, at best a simple relation between the given magnitudes. In contrast to that, macros are used in construction tasks. If the statement of a geometrical theorem is given (or hidden) in a construction, how can macros and modules be used when proving in geometry? The problem of argumentation and proof in geometry turns up - and will be treated in the next part of the paper. A more general view on proving and the use of pictures (incl. visualisation) can be found in Kadunz (2002).

4 Proof in Geometry: Change and Stability

At least one of the major aims of teaching and learning geometry is the exploration of non-trivial geometrical relations and the proof of their existence. Compared to other areas of school mathematics, the formulation of conjectures can be started earlier in geometry. For instance, an analysis of the remarkable points in triangles (like intersection of mid-perpendiculars, angle bisectors and the like) can be studied already in grade 6. This can be done using transformation geometry - even if arguments grounded in continuous movements and deformations seem to be more appropriate (see Bender 1989). From the study of these movements, insight in the generality of an argument can be developed, because the different configurations produced by the movement show a multitude of cases and even exceptional and limit cases. Exploration by using continuous movements may even generate new conjectures and ideas for proofs. In addition to that, they may visualize the flow of a proof by linking different stages of a proof (see Bender 1989, p. 129).

At the time Bender came up with these ideas, he was thinking about the use of films and changeable material models. Movements should not only link together individual steps of proofs but should also ease the restructuring of a drawing of a configuration. In order to show that all points with segments to the endpoints of a given line forming a right angle are situated on a circle with the given segment as diameter (the German "Thalessatz"), one can move the point to a position where the segments joining the endpoints of the given segment seem equally long (congruent). An immediate conjecture will be that there are isosceles triangles and the half of a square. Moving the point around could give rise to new observations and conjectures. If one compares the situation with traditional tools (when Bender was opting for movements) with the situation on DGS-use, the chances for the learners seem to be inverse: With DGS, movements and variation is not a problem because the dragmode of DGS allows for a variation of the drawing without changing the geometrical relations. What seems to be difficult for the learners with a DGS is the observation and exploration of a static situation, of a single representation. To come to a proof, the movement has to stop, the configuration has to be inspected and - eventually - arguments have to be brought forward. To find a proof, the learner has to see (well-)known geometrical properties and theorems ***into*** the configuration - whereas in the "olden" times of Bender, the movement which often could only be imagined in a thought experiment was an additive to the static representation and could then serve as support of the



Drawing 3

learning process. Consequently, the use of DGS and the dragmode needs a different support - namely a static picture of the geometrical statement or theorem. Unfortunately, pictures of geometrical statements are not part of a DGS. Nevertheless, the teacher can offer a set of pictures of geometrical statements which may support the development of proofs by the learner (see Kautschitsch 2001, Kadunz 2000).

The following problem may serve as an example: Let P be a point on a circle k with diameter AB. With s as the segment AP, one extends s in the direction of P so that a point Q is characterized by the property AP congruent PQ. Continuously changing P on circle k will also continuously change the position of Q. What about the locus of points of Q ?

The solution of this task may develop in two steps: At first , an appropriate drawing may be constructed and the locus of points Q may be observed. One immediately has a conjecture which in a second step should be proved. Using the locus-of-point feature of a DGS will soon lead to conjecturing that Q is moving on a circle with a diameter doubling the diameter AB of k. What about a proof? Students will again have to "read" the construction, i.e. they should use proven statements to show that the locus of points is a circle. The statements in the proof now take over the role of macros/modules in the construction task. In the construction, macros (incorporating elementary geometry statements) serve as operational units to build up the whole construction. For a proof, the entire argumentation has to be decomposed into argumentative units - and a list of prototypic pictures of geometrical statements may be of help. On the other hand, the dragmode may help to get an idea to start from. A decomposition of the configuration into flexible chunks could suggest some sort of "Thales-Theorem". The proof is then completed by an isosceles triangle (as can be seen in drawing 3). Within both steps of this proof, one could use pictures of the respective geometrical statements from a given list or individual or classroom experience to describe (parts of) the configuration. Nevertheless one should not forget that neither an individual drawing nor the DGS in use by themselves suggest the use of the respective and problem solving statements and/or pictures.

References

Bender, P. & Schreiber, A. (1985). Operative Genese der Geometrie. Stuttgart, Wien, B.G.Teubner, hpt.

Bender, P. (1989). Anschauliches Beweisen im Geomtrieunterricht - unter besonderer Berücksichtigung von (stetigen) Bewegungen bzw. Verformungen. Anschauliches Beweisen. Kautschitsch H. and Metzler W. (Eds). Stuttgart, Wien, B.G.Teubner, hpt: 95-146

Cypher, A. (Ed.) (1993). Watch what I do: programming by demonstration. Cambridge/Mass -London/UK, MIT Press.

Dörfler, W. (1991). Der Computer als kognitives Werkzeug und kognitives Medium. Computer - Mensch - Mathematik. Dörfler W. et al. (Eds.) Wien, Stuttgart, Hölder-Pichler-Tempsky; B.G.Teubner: 51-76.

Dubinsky, E. (1988). On Helping Students Construct the Concept of Quantification. 12th International Conference Psychology of Mathematics Education. Veszprem. vol. I: 255-262.

Graumann, G., R. Hölzl, et al. (1996). "Tendenzen der Geometriedidaktik der letzten 20 Jahre." Journal für Mathematikdidaktik 17(3/4): 163 - 237.

Kadunz, G. (2000). "Visualisierung, Bild und Metapher." Journal für Mathematik-Didaktik 21(3/4): 280-302.

Kadunz, G. (2002, in print). Visualisierung, die Verwendung von Bildern beim Lernen von Mathematik, Profil Verlag, München

Kastner, M. (2001). Geometrische Überlegungen zur Inversion am Kreis unter besonderer Berücksichtigung von dynamischer Geometriesoftware. Graduation at Inst.f.Mathematik, Univ. Klagenfurt.

Kautschitsch, H. (2001). DGS-unterstütztes Vermuten und Beweisen. Zeichnung - Figur - Zugfigur. H.-J. Elschenbroich, T. Gawlick and H.-W. Henn. (Eds) Hildesheim - Berlin, Franzbecker: 113-122

Lister, T. (1998). Hyperbolic Geometry using Cabri, (set of Cabri-macros available at http://mcs.open.ac.uk/tcl2/nonE/nonE.html).

Schweiger, F. (1992). "Fundamentale Ideen. Eine geistesgeschichtliche Studie zur Mathematikdidaktik." Journal für Didaktik der Mathematik (JMD) 13 (1992)(2/3): 199-214.

Sträßer, R. (2001). "Cabri-géomètre: Does a Dynamic Geometry Software (DGS) Change Geometry and its Teaching and Learning?" International Journal for Computers in Mathematics Learning 6(3): 319-333.

Sträßer, R. (2002, in print). "Makros" von DGS: Terme der Geometrie? Beiträge zum Mathematikunterricht 2002: Vorträge auf der 37. Bundestagung für Didaktik der Mathematik. W. Peschek et al. (Eds.) Bad Salzdetfurth, Franzbecker.

*The author thanks Rudolf Sträßer for constructive comments, suggestions and the transfer of his text from Austrian German into German English.

Author

Kadunz, Gert, Abteilung für Didaktik der Mathematik (ADM), Inst.f.Mathematik, Universität Klagenfurt, A-9020 Klagenfurt.
Email: gert.kadunz@uni-klu.ac.at