Adding Remote Computational Capabilities to Dynamic Geometry Systems

Jesús Escribano and Francisco Botana and Miguel A. Abánades

Departamento de Sistemas Informáticos y Computación, Universidad Complutense de Madrid, 28040 Madrid, Spain, escribano@sip.ucm.es
Departamento de Matemática Aplicada I, Universidad de Vigo, Campus A Xunqueira, 36005 Pontevedra, Spain, fbotana@uvigo.es
Ingeniería Técnica en Informática de Sistemas, CES Felipe II (UCM), 28300 Aranjuez, Spain, mabanades@cesfelipesegundo.com

Extended Abstract

A *Dynamic Geometry System* (DGS) is a computer application that allows the exact on-screen drawing of (generally) planar geometric constructions, and the dynamic manipulation of the construction by mouse dragging certain elements. The key property of these applications is the self adjustment of the position of all elements in the diagram to the changes produced by the user. This is also known as *Interactive* geometry. In this work, we study three DGS: *Cabri* [8], *The Geometer's Sketchpad* [6] and *Cinderella* [9].

From the beginning, DGS have been the paradigm of new technologies applied to Math education, area where they have found their most applications. Their convenience in the classroom is almost unanimously praised among education experts. However, questions have been raised on the influence of the use of DGS on the development of the concept of proof in school curricula [5]. Sometimes, formal proofs have been replaced by the construction of a great number of examples of a configuration, in what has come to be known as a *visual proof* (see the special issue in *Educational Studies in Mathematics* [7] for an extensive exposition on the subject).

This is a symptom of the incompleteness of general DGS relative to extra manipulation of configurations. In particular, out of the three DGS considered, only Cabri comes equipped with some property checker, and even in that case its numeric nature does not really provide a sound substitute for a formal proof.

To compensate the computational limitations of DGS, it seems clear that some symbolic capabilities have to be added to DGS. There are two main different approaches to implement this idea. Some systems incorporate their own code to perform symbolic computations (e.g. [4, 12]), while other systems choose to reuse existing *Computer Algebra Systems* (CAS) (e.g. [2, 10, 11]). Both strategies have been partially successful on addressing the three main issues in dynamic geometry, namely the continuity problem, the implementation of proof and discovery, and the determination of loci (see [1] for details). However, none of the main commercial DGS has tried to add ample symbolic capabilities in its latest versions. This is due in part to the very different nature of DGS and CAS. While

DGS are based on immediate interactive manipulations, CAS are based on question/answer queries. Besides these reasons, the main problem to connect existing DGS and CAS is the fact that different applications speak different languages. This is partly due to the lack of a *lingua franca* for *computer* Mathematics.

There are two main different, and somehow complementary, candidates to become the standard description of computational Mathematics, namely *MathML* ([13]), a W3C Recommendation oriented to Math presentation on the Web and *OpenMath* ([14]), a standard for representing mathematical objects together with their semantics.

We present an example of DGS-CAS communication between three of the most relevant commercial DGS (*Cabri*, *The Geometer's Sketchpad* and *Cinderella*) and the CAS *Mathematica* [15] and *CoCoA* [3], for which OpenMath has been chosen as the communicating language. The tool is a web application designed to symbolically process locus, proof and discovery tasks on geometric diagrams constructed with the considered DGS. Named **LAD** (acronym for *Locus-Assertion-Discovery*), it can be seen as a remote *add-on* for the three considered DGS. In other examples of connections of commercial DGS with commercial CAS, the user is required to have the appropriate (expensive) CAS as well as a specific software installed in the local drive of the user's computer, making process too burdensome for a standard DGS user. In our case, to use LAD the user is only required to have access to his/her favourite DGS and an internet connection.

From the DGS internal textual representation of a geometric diagram, an OpenMath description of the requested task is created using the elements in the *plangeo* OpenMath *content dictionaries*. Once the computational task (locus, assertion or discovery) has been determined by the user in the applet menu, a final description of the task is produced in terms of points and point constraints. The final computations follow the Groebner using CoCoA and the Mathematica kernel.

Moreover, **LADucation**, a *one-click* educational version of LAD designed as a way to work the idea of mathematical proof within a *Math lab*, is also presented. By just uploading the file generated by the considered DGS, LADucation computes graphs and equations of geometric loci. These equations make for mathematical sound proofs that complement the graphing abilities of the considered DGS. Besides computing geometric loci, in the case of a Cabri geometric construction, LADucation also gives *certified* answers to the five questions considered by the Cabri in-menu property checker.

LAD and LADucation are freely available at

`http://nash.sip.ucm.es/LAD/LAD.html`
`http://nash.sip.ucm.es/LAD/LADucation.html`

where detailed instructions can also be found.

Acknowledgments.

This work has been partially supported by UCM research group ACEIA and research grants MTM2004-03175 (Botana) and MTM2005- 02865 (Abánades, Escribano) from the Spanish MEC.

References

1. Botana, F., Recio, T.: Towards solving the dynamic geometry bottleneck via a symbolic approach. *Proc. 5th Int. Workshop on Automated Deduction in Geometry* (ADG 2004), LNAI, 3763, 92–110 (2006)
2. Botana, F., Valcarce, J.L.: A dynamic-symbolic interface for geometric theorem discovery. *Computers and Education*, 38(1–3), 21–35 (2002)
3. Capani, A., Niesi, G., Robbiano, L.: CoCoA, a system for doing Computations in Commutative Algebra. Available via anonymous ftp from: `cocoa.dima.unige.it`
4. Gao, X.S., Zhang, J.Z., Chou, S.C.: *Geometry Expert*. Nine Chapters, Taiwan (1998)
5. Hoyles, C., Jones, K.: Proof in Dynamic Geometry Contexts. Perspectives on the teaching of Geometry for the 21st Century. Kluwer (1998) 121-128
6. Jackiw, N.: *The Geometer's Sketchpad v 4.0*. Key Curriculum Press, Berkeley (2002)
7. Jones K., Gutirrez A., Mariotti M.(Guest Editors): Proof in Dynamic Geometry Environments: A PME Special Issue. Educational Studies in Mathematics **44** (2004).
8. Laborde, J. M., Bellemain, F.: *Cabri Geometry II*. Texas Instruments, Dallas (1998)
9. Richter–Gebert, J., Kortenkamp, U.: *The Interactive Geometry Software Cinderella*. Springer, Berlin (1999)
10. Roanes–Lozano, E., Roanes–Macías, E., Villar, M.: A bridge between dynamic geometry and computer algebra. Mathematical and Computer Modelling **37**(9–10) (2003) 1005-1028 .
11. Wang, D.: GEOTHER: A geometry theorem prover. *Proc. 13th International Conference on Automated Deduction* (CADE 1996), LNCS, 1104, 166–170 (1996)
12. `http://www.geogebra.at`
13. `http://www.w3.org/Math/`
14. `http://www.openmath.org/`
15. `http://www.wolfram.com/`