

Adaptative Learning Environment for Geometry

Vanda Santos
*CISUC, University of Coimbra
Portugal*

Pedro Quaresma
*CISUC/Department of Mathematics, University of Coimbra
Portugal*

1. Introduction

Geometry with its formal, logical and spatial properties is well suited to be taught in an environment that includes dynamic geometry software (DGSs), automatic theorem provers (ATPs) and repositories of geometric problems (RGP). With the integration of those tools in a given learning management system (LMS), we can build an environment where the student is able to explore the built-in knowledge, but also to do new constructions, and new conjectures, allowing, in this way, a better understanding of the concepts presented in a given e-course.

Dynamic geometry software are already well know tools for their educational properties, (e.g., CINDERELLA, GEOMETER'S SKETCHPAD, CABRI, GCLC, EUKLEIDES), with them we can visualize geometric objects and link formal, axiomatic nature of geometry with its standard models and corresponding illustrations, e.g., Euclidean Geometry and the Cartesian model. The common experience is that dynamic geometry software significantly help students to acquire knowledge about geometric objects. The visualization and the possibility of dynamically modify some of the parameters of a given geometric construction, are a very important tool in the comprehension of the geometric problems.

Automated theorem provers are less known as tools used in a learning environment, but geometry with its axiomatic nature is a "natural" field for a formal tool such as the ATPs. Automated theorem proving in geometry has two major lines of research: synthetic proof style and algebraic proof style (Matsuda & Vanlehn, 2004). Algebraic proof style methods are based on reducing geometric properties to algebraic properties expressed in terms of Cartesian coordinates. Synthetic methods attempt to automate traditional geometry proof methods. The synthetic methods (e.g. the Area Method (Chou et al., 1996a; Quaresma & Janičić, 2006a; Zhang et al., 1995)) provide traditional (not coordinate-based), human-readable proofs. In either cases (algebraic or synthetic) we claim that the ATPs can be used in the learning process (Jančić & Quaresma, 2007; Quaresma & Jančić, 2006b), and we will elaborate on this on the main body of this text.

An important first integration of tools is given by the DGSs incorporating ATPs, e.g. the GCLC DGS/ATP (Jančić & Quaresma, 2006). With a DGS we can visualize a given geometric construction, we can also dynamically change some objects of the construction and test, in this way, if some property holds in all checked cases, but this still does not prove that the construction is sound. The integration of a given ATP in a dynamic geometry tool allow the

use of the prover to reason about a given DGS construction, this is no longer a “proof by testing”, but an actual formal proof. This integration allow the use of the DGS construction in the ATP without changing and adapting it for the deduction process, adding the conjecture to be proved is the only step needed. If the ATP is able to produce synthetic proofs, the proof itself can be an object of study, in the other cases only the conclusion matters (Chou et al., 1996c; Janičić & Quaresma, 2006; Quaresma & Janičić, 2006a).

Another link between the ATPs and the DGSs is given by the automated deductive testing, by the ATP, of the soundness of the constructions made by the DGS (Jančić & Quaresma, 2007). Most, if not all DGSs are able to detect and reports syntactic, and semantic errors, but the verification of the soundness of the construction is beyond their capabilities. If we have this kind of integration between DGSs and ATPs we can check the soundness of a given construction, and not only its syntactic and semantic correction. Again if the ATP produces synthetic proofs, the proof itself can be an object of study, providing a logical explanation for the error in the construction.

Repositories of geometric problems are important as a source of geometric constructions, figures, problems and proofs, which can be used for teaching and studying geometry. Teachers will appreciate the available information, to build new teaching examples, quizzes and tests. Students will appreciate the source of different examples to complement the information already given in class. Many of the DGSs available already contain a set of examples in their distributions, but by repositories of geometric problems we mean a publicly accessible and Internet based database of geometric constructions, figures and proofs. That is, a Web repositories where a mathematician can go and browse, download, upload geometric problems. Such a Web repository should be independent of any DGS/ATP, allowing its information to be used for a large array of such systems. The system GEOTHMS (Quaresma & Jančić, 2007) already fulfill some of this requirements.

The Learning Management Systems (LMS) (e.g. MOODLE, CLAROLINE, WEBCT) are systems that were made to support the learning process, most of them allow the learning with the extensive use of electronics means, without physical boundaries. With the LMSs it is possible to create an asynchronous (but also with some synchronous features), collaborative environment, where the learning by experience can be blended with other forms of learning.

To build what we call an *adaptive learning environment for geometry*, we have to integrate the DGSs the ATPs and a repository of problems in a LMS in order to build an environment where the student can have a broad experimental, but with a strong formal support, learning platform.

Such an integration it is still to be done, there are already many excellent DGSs, some of them have some sort of integration with ATPs, others with RGP (Quaresma & Jančić, 2006b; Quaresma & Jančić, 2007), some attempts to integrate this tools in a LMS have already been done (Santos & Quaresma, 2008), but, as far as we know, all this integrations are partial integrations. A system where all this tools are integrated and can be used in a fruitful fashion does not exist yet.

In this text we describe our past, and present, experience in the integration of intelligent geometrical tools, and we explain our vision of an *adaptive learning environment for geometry*. A dynamic environment where the student can study the models and axiomatic theories of geometry, to understand the differences and connections between the two perspectives. An environment where the student should be challenged by new problems to be solved.

In the following sections we will write about the different software tools available for the “computational geometer”, their qualities from the view point of the learning process, first

each of the tools by themselves, then about their integration in one *adaptive learning environment for geometry*. In appendix we briefly review the tools presented, giving links to their Web home pages.

2. The Dynamic Geometry Software

The DGSs distinguish themselves from a drawing program in two major ways, the first is its knowledge of geometry, form a initial set of objects drawn freely in the Cartesian plane (or maybe another model of geometry), we specify/construct a given geometric figure using relations between the objects in the construction, e.g. the intersection of two non-parallel lines, a line perpendicular to a given line and containing a given point, etc. That is, when we use a DGS we are constructing a geometric figure with geometric objects and geometric relations between them. If it is true that a drawing program is almost useless in a learning environment for geometry, a DGS could be a very useful tool in such a learning environment.

One other major feature of a DGS is its capability to introduce dynamism to a given geometric construction. Given the fact that we specify/construct a geometric figure using a set of basic elements, e.g. points, lines, circles and relations between them, the DGSs allow its user to move one of the basic elements from its initial placement to another placement in the Cartesian plane, the relations will be kept, so a movement in a single point can entail the movement of almost all the other elements in the construction, i.e. when moving a basic object, we will move that object and all the other elements that are related to it.

Using the dynamism of a DGSs we can illustrate some geometric property, giving a strong evidence (not a proof) that the property in question will hold. For example the following geometric property (example 85 in Chou (1987)).

Theorem 1. *Let P and Q be two points on side BC and AD of a parallelogram $ABCD$, such that $PQ \parallel AB$, $M = AP \cap BQ$ and $N = DP \cap QC$. Then it can be shown that $MN \parallel AD$ and $MN = AD/2$.*

The construction made using a given DGS should respect the fact that the only free points are the points A , B , C and D , all the other points are constructed points, e.g. P and Q are points belonging to two lines, and such that PQ is parallel to another line.

Given that we can see that the property $MN \parallel AD$ (seems to) hold. Using the dynamic capabilities of the DGSs we can show that moving around any of the free points does not modify that property of the geometric construction (see Fig. 1). This is not a proof and that fact should be stressed by a teacher using this kind of examples, but nevertheless, it is an appealing capability that can be used even to introduce the subject of rigorous proofs in geometry.



Fig. 1. Example 85 in Chou (1987) in Cinderella2

Another possible use of the dynamism of the DGS tools is given by some visual proofs, e.g. Pythagoras theorem has many very beautiful visual proofs. For example in Fig.2 we have 10 snapshots of a visual proof of the Pythagoras theorem made using JGEX. This is one of the examples contained in the distribution.

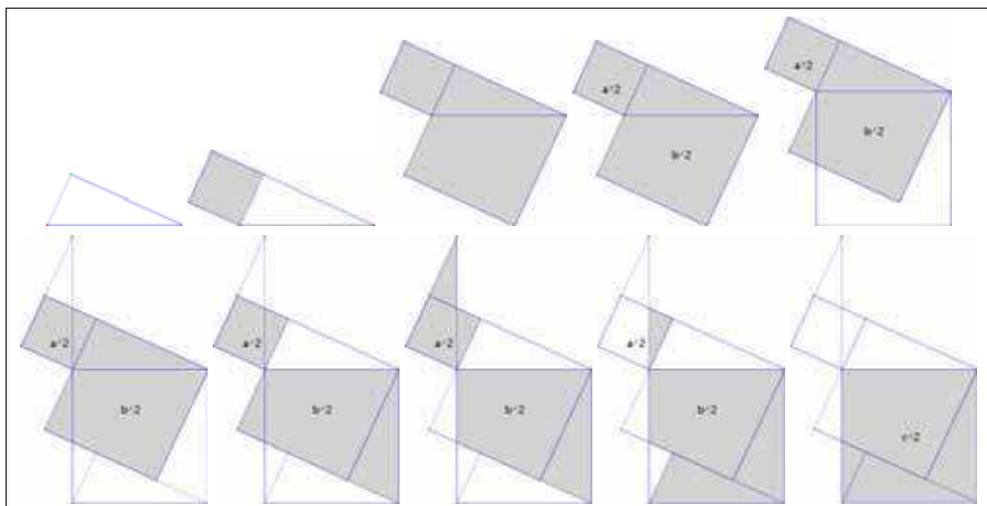


Fig. 2. Pythagoras Theorem - Visual Proof

This visual proof is animated, when we load this example we can see an animated sequence of steps, from the initial construction to the final one, and with the sliding of the different components from one place to the other. This kind of feature can be used in a very fruitful way in a learning situation.

3. Automatic Theorem Proving in Geometry

As seen above the powerful graphical interface of a DGS allow its users to explore freely the geometric models, they are very important tools in a learning environment for geometry. But it is also true that they can only be used as a support, a first step, in a more formal reasoning about a given geometrical construction. To prove a given result using a DGS we would have to prove that the result is true in every possible configurations, in most of the times this is not possible. Fortunately we begin to have at our disposal tools to reason about geometrical constructions, the geometrical ATPs.

To introduce the deductive reasoning, and in particular the use of automatic theorem proving tools, in a helpful way to the mathematical student, the Euclidean geometry is a good area of study to begin with. Geometry has a simple set of axioms, and a simple set of deduction rules, easy to understand, yet very powerful is its reach. Euclidean geometry has also an easy connection with its standard models, providing a visual counterpart to some of its proofs. Looking the other way around if we want to study the formal methods of geometry the geometric ATPs are already well developed tools.

The different ATP methods and tools that can be used in an learning environment, have different fields of application. As already said automated theorem proving in geometry has two major lines of research: synthetic proof style and algebraic proof style.

The algebraic proofs are, in our opinion, not very useful if we want to study/teach the deductive method in geometry, the ATPs of this type begin by reducing the geometric properties to algebraic properties expressed in terms of Cartesian coordinates, making the proof by pure algebraic methods. The student gains no insight about the geometrical result by looking into this type of proofs. They can be used, and are already being used, as a support tool, e.g. when we want to prove some geometric conjecture without caring about the proof itself.

Synthetic methods attempt to automate traditional geometry proof producing human-readable proofs. We can sub-divide the ATPs in this area in those that try to automate the traditional geometry proof methods focusing in some specific type of problems, and using some heuristics to try to control the space solutions explosion (Coelho & Pereira, 1986); and those based in the area method (Chou et al., 1996a;b; Quaresma & Janičić, 2006a). Both this types are suitable for our purpose of building an e-course for learning formal reasoning in geometry, in the following we will focus in the ATP that one of the authors worked on, the GCLC_{PROVER} (Janičić & Quaresma, 2006; Quaresma & Janičić, 2006a), which is based in the area method.

In either case the geometrical ATPs can be used, in a very fruitful way, in the learning process (Janičić & Quaresma, 2007; Quaresma & Janičić, 2006b). In the following we will present two different uses of the ATPs in geometry, showing some examples of application of existing software tools.

3.1 Construction Validation

As already said the DGSs visualize geometric objects and link formal, axiomatic nature of geometry (most often — Euclidean) with its standard models (e.g., Cartesian model) and corresponding illustrations.

Some of the DGSs have a sort of dual view of a given geometrical construction, a formal language describing it, and a graphical interface where the construction given by the formal description is draw (e.g. EUKLEIDES, GCLC), others do not possess, at least in an explicit form, a formal language for geometric constructions, instead they have a graphical interface where the user can “draw” the geometric constructions, using a pre-defined set of geometric tools (e.g. CINDERELLA, GEOMETER’S SKETCHPAD, CABRI).

In either cases we have three types of construction errors:

- syntactic errors - for a DGS with a formal language this type of error is easily detected by its processor, for the others this type of error doesn’t occur given the fact that the user is in a controlled environment where only syntactically corrected actions are allowed. In either cases this are the least important, an easily correctable, errors.
- semantic errors - situations when, for a given concrete set of geometrical objects, a construction is not possible. For instance, two identical points do not determine a line, this type of error will be dealt by most (if not all) of the DGSs for a given fixed set of points e.g. it is an argument only suited for a given instance (in the model, e.g. the Cartesian plane) of the construction.
- deductive errors - a construction geometrically unsound, e.g., the intersection of two parallel lines in Euclidean geometry. The formal proof of the soundness (or not) of a given construction can only be dealt by DGSs that incorporate an ATP capable of geometric reasoning.

Again (see section 2) we use the example 85 in Chou (1987) to illustrate this.

In EUKLEIDES we can construct this figure with the specification presented in Fig. 3.

```

% building commands - Chou's book - example 85
frame(0,0,11,17)
A = point(1,1); B = point(7,1); C = point(10,6)
ab = line(A,B); bc = line(B,C)
ad = parallel(bc,A); cd = parallel(ab,C)
D = intersection(ad,cd)
P = point(bc,1.5)
pq = parallel(ab,P)
Q = intersection(pq,ad)
pd = line(P,D); qc = line(Q,C)
N = intersection(pd,qc)
pa = line(P,A); qb = line(Q,B)
M = intersection(pa,qb)
% drawing commands
draw(M);label(M,90:); draw(N);label(N,90:)
draw(A);label(A,-90:); draw(B);label(B,-90:);
draw(C);label(C,90:)
draw(D);label(D,90:); draw(P);label(P,0:);
draw(Q);label(Q,180:)
draw(segment(P,D)); draw(segment(P,A))
draw(segment(Q,C)); draw(segment(B,Q))
draw(segment(P,Q)); draw(segment(A,B))
draw(segment(A,D)); draw(segment(C,B))
draw(segment(C,D)); draw(line(M,N),dashed)

```

Fig. 3. Eukleides Code for Example 85 of Chou's Book

Which gives the illustration of Fig. 4.

If we add the code for the intersection of lines AD and MN , e.g.

```

mn = line(M,N)
X = intersection(mn,ad)

```

EUKLEIDES will not perform the offending step and it will give the following error message:

```
Error at line 22: parallel lines.
```

This is semantic only warning, given for this set of points in the Cartesian plane, the same type of warning is given by WINGCLC:

```
Run-time error: Bad definition. Can not determine intersection.
(Line: 40, position: 10)
```

Most of the DGSs without a formal language will not allow their users to perform the illegal construction, e.g. CINDERELA2 doesn't perform the illegal construction but is unable to say why.

A more interesting approach is given by the DGSs that incorporate an ATP for geometry. For example the same WINGCLC already mentioned above is capable, if called with an appropriated option, of interaction with the built-in ATP, GCLCPROVER, giving the following justification:

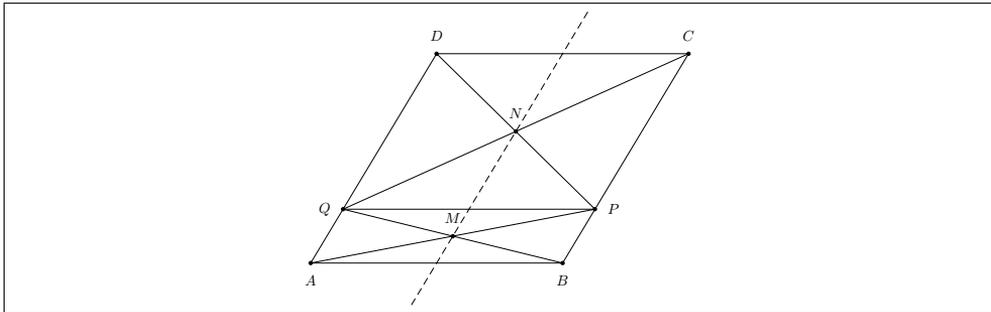


Fig. 4. Example 85 of Chou’s Book

Deduction check invoked: the property that led to the error will be tested for validity.

The conjecture successfully proved - the critical property always holds. The prover output is written in the file error-proof.tex.

and a WINGCLC user will have, not only the statement that the construction is always unsound, but also a proof of it in the area method style. The proof is a synthetic proofs, a geometric proof that can be easily (unfortunately it is not always the case) read by mathematicians. A similar approach is given by the DGS/ATP JGEX, during the construction it calculate fix-points of the figure, so, when a user tries to perform an illegal construction, the tool is capable of saying why it is not possible to perform the construction. Unfortunately it doesn’t provide a formal proof of its reasoning (see Fig. 5).

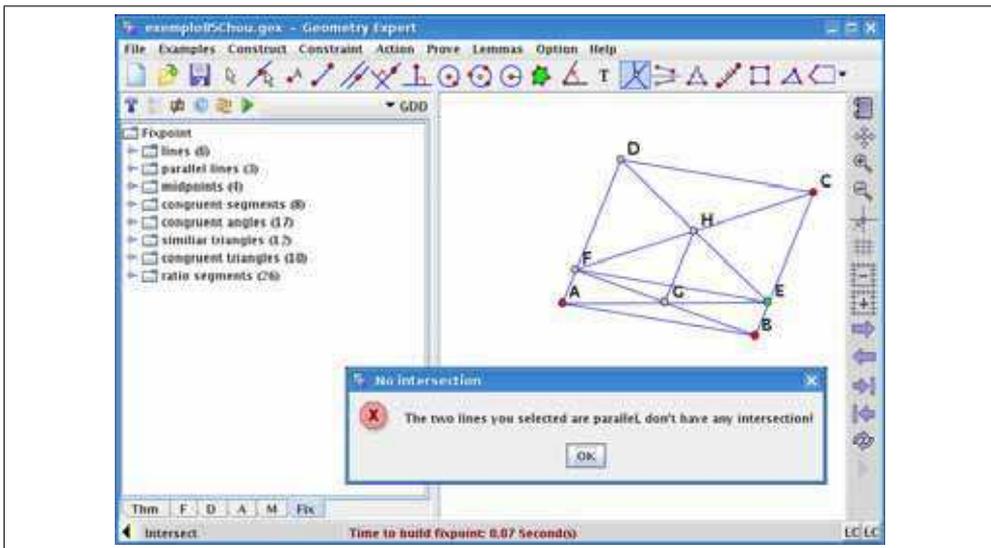


Fig. 5. JGEX Construction Validation

As seen on this section there are already some DGSs that, integrating ATPs for geometry, are capable of performing deductive checks, stating why a given construction is unsound and, in the case of GCLC, even to provide a proof of this fact.

This capability of performing deductive checks is, in our opinion, a very interesting and important feature. It enhances the didactic nature of the DGSs providing a first link between these tools and the ATPs, a link between the model of geometry the construction is based on, and the geometric theory behind that model.

3.2 The Area Method

This method is a synthetic method providing traditional (not coordinate-based), human-readable proofs (Chou et al., 1993; 1996a; Narboux, 2004; Quaresma & Janičić, 2006a). The proofs are expressed in terms of higher-level geometric lemmas and expression simplifications. The main idea of the method is to express hypotheses of a theorem using a set of constructive statements, each of them introducing a new point, and to express a conclusion by an equality of expressions in geometric quantities, without referring to Cartesian coordinates. The proof is then based on eliminating (in reverse order) the points introduced before, using for that purpose a set of appropriate lemmas. After eliminating all introduced points, the current goal becomes an equality between two expressions in quantities over independent points. If it is trivially true, then the original conjecture was proved valid, if it is trivially false, then the conjecture was proved invalid, otherwise, the conjecture has been neither proved nor disproved. In all stages, different simplifications are applied to the current goal. Some steps require proving some lemmas (giving proofs on different levels).

The basic geometric quantities used in this method are *ratio of directed segments* ($\frac{\overline{AB}}{\overline{CD}}$), *signed area* (S_{ABC} , signed area of a triangle ABC) and *Pythagoras difference* ($P_{ABC} = AB^2 + CB^2 - AC^2$) (for details see Quaresma & Janičić, 2006a). The conjecture is built from these geometric quantities (over points already introduced within the current construction), eventually combined together by standard arithmetic operators. A wide range of geometric conjectures can be simply stated in this way, e.g.:

- points A and B are identical;
- points A, B, C are collinear;
- AB is perpendicular to CD ;
- AB is parallel to CD ;
- O is the midpoint of AB ;
- AB has the same length as CD ;
- points A, B, C and D are harmonic.

To our knowledge, apart from the original implementation by the authors who first proposed the method (Chou et al., 1993; 1996a), there are other three implementations more, all of them were made independently and in different contexts.

- within a dynamic geometry tool GEX (now JGEX), implemented by the original authors (Chou et al., 1996b;c).
- within a dynamic geometry tool (GCLC (Janičić, 2006)), implemented by Predrag Janičić and Pedro Quaresma (Janičić & Quaresma, 2006);
- within a generic proof assistant (COQ (Team, 2007)), implemented by Julien Narboux (Narboux, 2004);

- within a tool for storing and exploring mathematical knowledge (Theorema (Buchberger et al., 2006)), implemented by Judite Robu (Robu, 2002).

From this four the GCLC_{PROVER} theorem prover excel in the learning area, given that it can produces proofs (in \LaTeX form) that are readable by mathematicians. Some of the proofs produce by GCLC_{PROVER} can be very short and elegant.

For example, the proof of Midpoint Theorem (example 6.39 in Chou et al., 1994), requires only 45 steps and takes 0.001 seconds.

Theorem 2 (Midpoint Theorem). *Let ABC be a triangle, and let A' and B' be the midpoints of AC and BC respectively. Then the line A_1B_1 is parallel to the base AB .*

We can use GCLC to produce these geometric construction (see Fig. 6), we have to define three points A , B and C , and from this ones all the other objects should be constructed, namely the points A_1 and B_1 .

```

point A 5 5
point B 45 5
point C 20 30

midpoint B_1 B C
midpoint A_1 A C

drawsegment A B
drawsegment A C
drawsegment B C
drawsegment A_1 B_1
cmark_b A
cmark_b B
cmark_t C
cmark_lt A_1
cmark_rt B_1

prove {equal {signed_area3 A_1 B_1 A} {signed_area3 A_1 B_1
B}}
```

Fig. 6. Midpoint's Theorem in GCLC language

As we can see the conjecture $AB \parallel A_1B_1$ could be expressed in the geometric quantities used in the area method, in this case we have $S_{A_1B_1A} = S_{A_1B_1B}$, and the conjecture is incorporated in the GCLC code.

The proof, as already said, is very short and easily read by a mathematician (see Fig. 7). Unfortunately not every proof produced by this ATP is this simple, but a clever choice of examples can be used to show the deductive reasoning in geometry and in this way to improve the students awareness about the links between the models of geometry and its theories.

4. Repositories of Geometric Problems

Repositories of geometric problems are important as a source of geometric constructions, figures, problems and proofs, which can be used for teaching and studying geometry.

(1)	$S_{A_i B_i A} = S_{A_i B_i B}$, by the statement
(2)	$S_{B_i A A_i} = S_{B_i B A_i}$, by geometric simplifications
(3)	$\left(S_{B_i A A} + \left(\frac{1}{2} \cdot (S_{B_i A C} + (-1 \cdot S_{B_i A A})) \right) \right) = S_{B_i B A_i}$, by Lemma 29 (point A_i eliminated)
(4)	$\left(0 + \left(\frac{1}{2} \cdot (S_{B_i A C} + (-1 \cdot 0)) \right) \right) = S_{B_i B A_i}$, by geometric simplifications
(5)	$\left(\frac{1}{2} \cdot S_{B_i A C} \right) = S_{B_i B A_i}$, by algebraic simplifications
(6)	$\left(\frac{1}{2} \cdot S_{B_i A C} \right) = \left(S_{B_i B A} + \left(\frac{1}{2} \cdot (S_{B_i B C} + (-1 \cdot S_{B_i B A})) \right) \right)$, by Lemma 29 (point A_i eliminated)
(7)	$\left(\frac{1}{2} \cdot S_{A C B_i} \right) = \left(S_{B A B_i} + \left(\frac{1}{2} \cdot (S_{B C B_i} + (-1 \cdot S_{B A B_i})) \right) \right)$, by geometric simplifications
(8)	$S_{A C B_i} = (S_{B A B_i} + S_{B C B_i})$, by algebraic simplifications
(9)	$\left(S_{A C B} + \left(\frac{1}{2} \cdot (S_{A C C} + (-1 \cdot S_{A C B})) \right) \right) = (S_{B A B_i} + S_{B C B_i})$, by Lemma 29 (point B_i eliminated)
(10)	$\left(S_{A C B} + \left(\frac{1}{2} \cdot (0 + (-1 \cdot S_{A C B})) \right) \right) = (S_{B A B_i} + S_{B C B_i})$, by geometric simplifications
(11)	$\left(\frac{1}{2} \cdot S_{A C B} \right) = (S_{B A B_i} + S_{B C B_i})$, by algebraic simplifications
(12)	$\left(\frac{1}{2} \cdot S_{A C B} \right) = \left(\left(S_{B A B} + \left(\frac{1}{2} \cdot (S_{B A C} + (-1 \cdot S_{B A B})) \right) \right) + S_{B C B_i} \right)$, by Lemma 29 (point B_i eliminated)
(13)	$\left(\frac{1}{2} \cdot S_{A C B} \right) = \left(\left(0 + \left(\frac{1}{2} \cdot (S_{A C B} + (-1 \cdot 0)) \right) \right) + S_{B C B_i} \right)$, by geometric simplifications
(14)	$0 = S_{B C B_i}$, by algebraic simplifications
(15)	$0 = \left(S_{B C B} + \left(\frac{1}{2} \cdot (S_{B C C} + (-1 \cdot S_{B C B})) \right) \right)$, by Lemma 29 (point B_i eliminated)
(16)	$0 = \left(0 + \left(\frac{1}{2} \cdot (0 + (-1 \cdot 0)) \right) \right)$, by geometric simplifications
(17)	$0 = 0$, by algebraic simplifications

Q.E.D.

There are no ndg conditions.
 Number of elimination proof steps: 5
 Number of geometric proof steps: 15
 Number of algebraic proof steps: 25
 Total number of proof steps: 45
 Time spent by the prover: 0.001 seconds

Fig. 7. Proof of the Midpoint Theorem

Many of the DGSs available contain a set of examples, but by repositories of geometric problems we mean a publicly accessible and Internet based database of geometric constructions, figures and proofs. That is, a Web repositories where a mathematician can go, browse the list of existing problems, and download and/or upload geometric problems.

In a limited way the Web-site GEOMETRIAGON, is an example of such a repository of geometric objects. The GEOMETRIAGON has an already vast repository of problems in the area of classical constructive (ruler and compass only) Euclidean geometry (see Fig. 8).

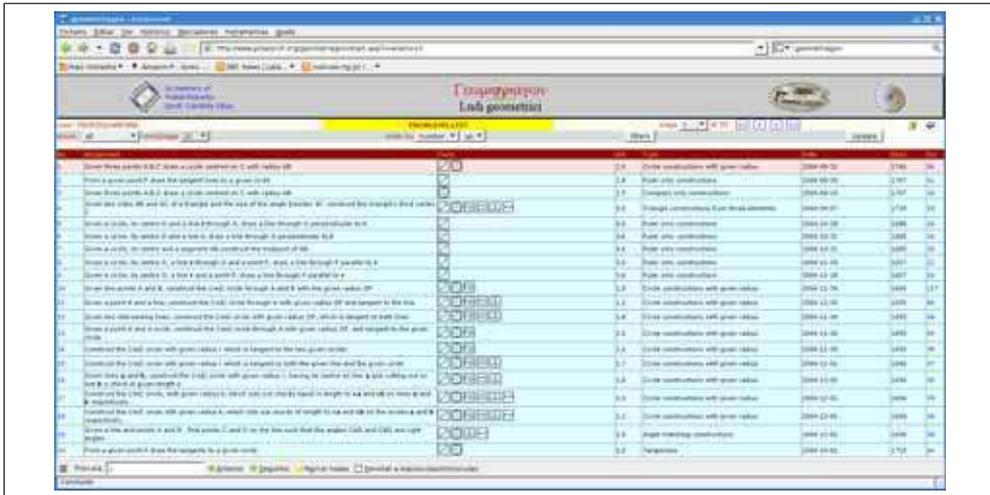


Fig. 8. Geometriagon List of Problems

A registered user can access/edit all problems and solutions. It incorporate a clever java applet where a user, for a chosen problem, can perform only valid steps in a geometric construction, using only a limited set of tools. The system is capable to recognize whenever a user has reach a solution of a problem (see Fig. 9). Notice that this page is in Portuguese, the GEOMETRIAGON Web page has an interface in seven different languages, this internationalization feature is interesting when we think in terms of high-school education, where an interface in a native language is much more appealing then in a foreign language.

The GEOMETRIAGON does not provide any type of output, neither for the proofs, neither for the constructions, it is a close solution. It does not provide any links with ATPs.

A Web-system that was thought, from the start, as a Web-repository for geometric problems is the GEOTHMS system. The GEOTHMS system is a Web-based laboratory for exploring geometrical knowledge that integrates DGSs, ATPs, and a repository of geometrical constructions, figures and proofs (see Fig.10).

A GEOTHMS user can browse through a list of available geometric problems, their statements, illustrations, and proofs. The GEOTHMS user can also interactively produce new geometrical constructions, theorems, and proofs and add new results to the existing ones. In its current status (June, 2009) the list contains 127 problems (see Fig. 11).

The GEOTHMS system integrates two DGSs, EUKLEIDES and GCLC and two ATPs, GCLCPRover and COQAREAMETHOD, allowing its user to use this tools and the list of prob-

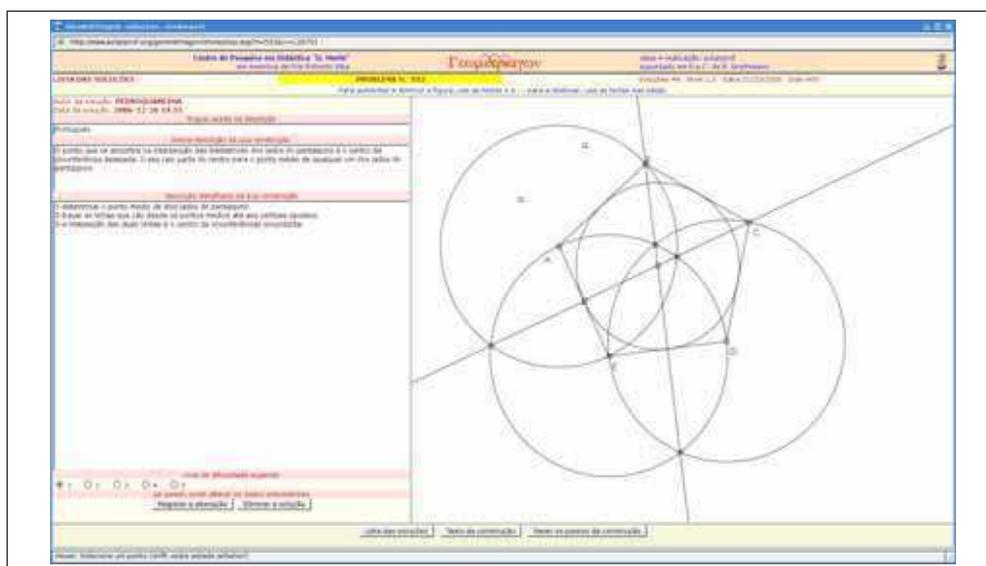


Fig. 9. Geometriagon Solution Solver Applet

lems in a integrated environment. In a next section we will return to this subject of tools integration.

5. Learning Management Systems

The Learning Management Systems (LMSs) are systems that are made to support the learning process, in most of them we can have:

- an extensive use of computers in the learning process;
- learning without physical boundaries (through Web interfaces);
- learning by experience;
- learning in a collaborative environment;
- learning in asynchronous environment, having also some synchronous features.

This type of tool make possible the administrative tasks, the pedagogical support, the management and distribution of contents to the students, as well as the interactivity between all the participants (teachers, students). Given what was said we felt as an important feature of a learning environment in geometry to be able to use all the types of geometric software already described, the LMS as a tool to give support to a Web/asynchronous (or synchronous)/collaborative learning environment, the DGSs and ATPs as tools to strengthen the learning by experience counterpart, and the RGP as a source of examples, conjectures, proofs, to be used by the DGSs and ATPs.

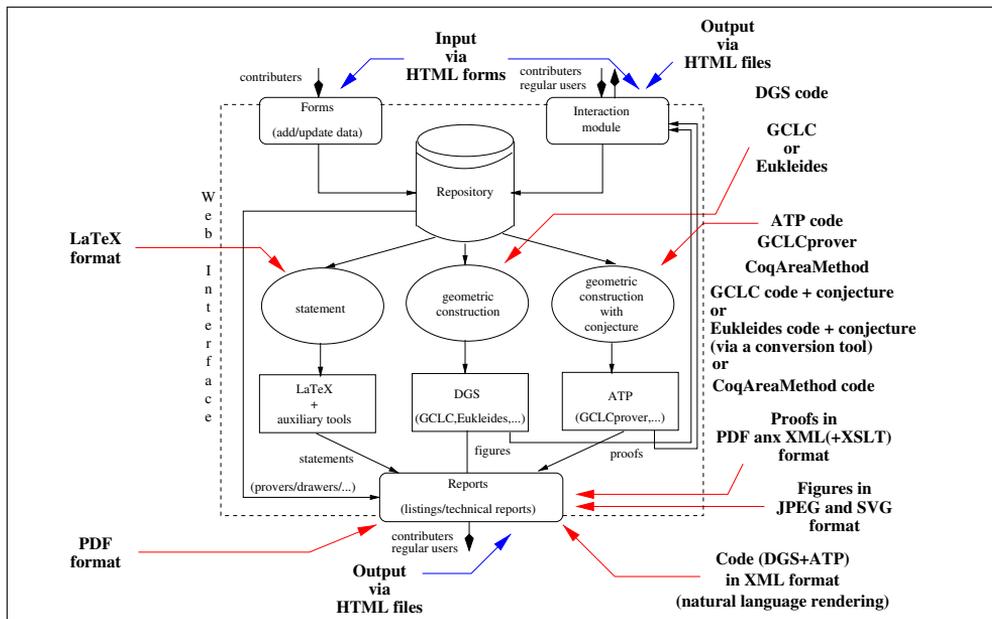


Fig. 10. GeoThms Layout

6. Tools Integration

As said in the previous sections the DGSs allow its user to explore the models of geometry putting in practice the popular saying “a picture is worth a thousand words”, the ATPs are very important because they allow its users to explore the axiomatic nature of geometry, as said geometry has a simple set of axioms, and a simple set of deduction rules, easy to understand, yet very powerful is its reach. The LMS support the learning process, they allow the learning with the extensive use of electronics means, without physical boundaries and asynchronously. Finally the repositories of problems are important as a source of information to be used in the DGSs, ATPs and/or in a e-course implemented in a LMS.

Given this, we feel as very important the integration of this tools in one integrated learning environment, where the LMS could be used as a tool to give support to a Web/asynchronous (or synchronous)/ collaborative learning environment, and the links to the DGSs, ATPs, and repositories of problems would be used to strengthen the learning by experience counterpart. The integration of this tools is already partially made by many of the existing systems. Most, if not all, the DGSs have a list of examples available to their users (and only those), there are already some DGSs with some type of integration with ATPs, e.g. JGEX, GCLC, GEOPROOF, also worth mention the DGSs CINDERELLA with its use of randomized theorem for checking geometrical properties. The integration with Web-repositories and with LMS is more scarce, as far as we know the GEOTHMS is the unique system that integrates DGSs (two), ATPs (two) and a repository of problems widely accessible via a Web-page. The explicit integration with the LMSs is also almost undone.

Name	Short Description	Figure	N. Figures	N. Proofs	See details
Cleidwell - Example 3.3	On the two sides AC and BC of triangle ABC, two squares ACDE and BCFG are drawn. M is the midpoint of AB. Then it can be showed that $DM \perp CG$. (Cleidwell, pg 82)		1	0	See details
Circumcenter of a Triangle			1	1	See details
Desargues' Theorem			1	2	See details
Distance of a line containing the centroid to the vertices			1	1	See details
Euler's Line			1	0	See details
Example 2.8, Cleidwell, pg 88	Let ABC be a triangle such that $AC=BC$. D is a point on AC. E is a point on BC such that $AD=BE$. If F is the intersection of DE and AB then $DF=EF$.		1	1	See details
Example 3.8, Cleidwell	Three equilateral triangles A ₁ BC, AB ₁ C and ABC ₁ are erected on the three respective sides of BC, CA, and AB of a triangle ABC, then the lines AA ₁ , BB ₁ and CC ₁ are concurrent.		1	0	See details
Gauss Point Theorem	Given five lines, we have five Gauss lines. These five Gauss lines are concurrent. The point of concurrency is called the Gauss point. Examples 18, Cleidwell, pg 109		1	1	See details
Gauss line Theorem			1	1	See details

Fig. 11. GeoThms: List of Problems

In the GEOThMS system (one of the authors is also one of the authors of that system) the integration was made via a XML-suite (Quaresma et al., 2008) a XML framework, that define a *normal form*, allowing linking the different formats used by the DGSs and ATPs.

These are some of the most important motivating arguments for using XML in storing descriptions of geometrical constructions and proofs, and as interchange format:

- instead of raw, plain text representation, geometrical constructions will be stored in strictly structured files; these files will be easy to parse, process, and convert into different forms and formats;
- input/output tasks will be supported by generic, external tools and different geometry tools will communicate easily;
- growing corpora of geometrical constructions will be unified and accessible to users of different geometry tools;
- easier communication and exchange of material with the rest of mathematical and computer science community;
- there is a wide and growing support for XML;
- different sorts of presentation (text form, \LaTeX form, HTML) easily enabled;
- strict content validation of documents with respect to given restrictions.

In the GEOThMS system there are already converters for the different tools incorporated in it, confirming, in this way, that the proposed XML format can serve its main purpose (see Fig. 12). This XML-suite could be used to make the different tools and their corpora, widely usable, helping in the wanted integration of tools in one single learning environment.

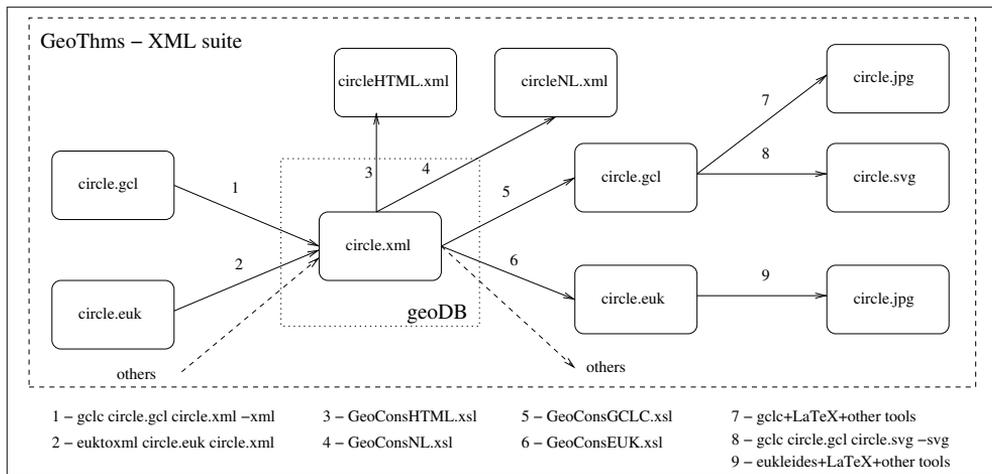


Fig. 12. GeoThms: XML Suite

The system GEOTHMS provides an Web-environment suitable for new ways of studying and teaching geometry at different levels, but it lacks the tools to support the learning process possess by the LMS systems. In (Santos & Quresma, 2008) the authors of this text describe a first attempt to integrate all this tools in a unique system. We have created a Web-page, (<http://hilbert.mat.uc.pt/GeoGCLC>), where all the objects and constructions available in GCLC are described in detail. This page also provides a workbench where a user can interactively use the GCLC.

The GEOTHMS system was also the base for an integration of the DGSs into the e-Learning course implemented in a LMS platform. The next section will describe this first effort in the direction of the final goal of an *adaptive learning environment for geometry*.

6.1 An e-Course in Geometry

In order to begin working in the direction of building an *adaptive learning environment for geometry*, an environment, build around an LMS, and integrating DGSs, ATPs, and RGP for geometry, we build an e-course, using a known LMS system, and integrating a DGS and a repository problems (Santos & Quresma, 2008). We will describe now this attempt.

The main goal of the e-Learning course in geometry was to introduce the Euclidean geometry as a formal discipline that, from a small set of objects: points, lines, circles, and also a small set of rules, can produce complex figures. A discipline where we can prove assumptions about the geometrical constructions, a discipline where we can reason about the created objects.

As a first step towards that goal we begin introducing the tool GCLC. To support the learning of the GCL syntax and semantics, in an e-Learning environment, we have created a Web page (<http://hilbert.mat.uc.pt/GeoGCLC>), where all the objects and constructions available in GGCL are described in detail. This page also provides a workbench where a user can interactively use the GCLC and it is the base for the integration of GCLC into the e-Learning course implemented in a LMS platform. (<http://hilbert.mat.uc.pt/Moodle/course/view.php?id=28>) (see Fig. 13).



Fig. 13. GCLC for Geometry e-Course in Moodle

The e-Learning course is divided in two main parts. The first one is, as already said, an introduction to the basic objects of Euclidean geometry and to the GCL syntax. In this section we introduce a glossary as a support object to the rest of the e-course.

The second part is all about constructions: from the basic ones to the more complex ones, ending with some notable results (e.g. the Ceva's theorem). For each of these constructions we begin defining the goal to be attained by the student and we also provide some initial information about the construction. After that we have some activities to be done by the student. We ask about the geometric figure to be produced and about the GCL code to realize it, the student is able to use the workbench in order to try to solve the purposed exercise (see Fig. 14). It is also possible to seek help given the fact that many constructions are already available in the workbench, that is, the actual figure, and the code to produce it, is accessible to the student. In the notable results some historical information is provided.

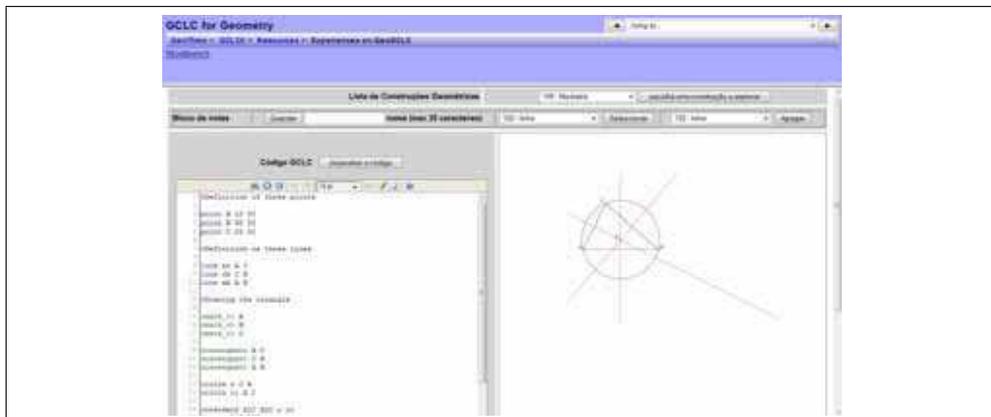


Fig. 14. GeoGCLC

In order to be independent of the implementation platform, and also be able to do an easy migration between different LMSs we implemented this e-course as a SCORM module (Wisher, 2009) (see Fig. 15).



Fig. 15. e-Course in SCORM norm

The SCORM elements can be easily combined with other compatibles elements to produce modular e-courses. The SCORM e-course implemented has all the necessary learning objects to provide a rich Web asynchronous environment, e.g. glossary, workbenches, lessons, tests and wikis, and it is independent of a given platform.

We can say that, at least in this first implementation, we could produce an DGS/RGP/LMS integrated learning environment in a specific LMS (Moodle), but also as a SCORM module, i.e. LMS independent.

7. Future Work

The integration of GEOGCLC, and via this one, the integration of GCLC, in an e-Learning environment course gives to the student in geometry a direct access to a DGS, creating in this way a workbench where the student can explore the constructions already built-in, to transform them, and even to create new ones keeping all the constructions in a personal folder. In this way we provide a strong contribution to the “learning by experience” component of an e-Learning course. The GCLC tool integrates also an automatic theorem prover (ATP) based on the area method, the same type of integration can be done with the system GEOTHMS. Such an e-Learning course will give the student an opportunity to use the ATP to prove conjectures about the construction made, i.e., providing an e-Learning environment to study formal proofs in Euclidean Geometry.

One project that the authors of this text are working on is to build an *adaptative learning environment for geometry*. A dynamic environment where the student can study the models and axiomatic theories of geometry, to understand the differences and connections between the two perspectives. An environment where the student should be challenged by new problems to be solved by the student.

We intend to do this in steps. First we want to implement a LMS module which should allow the use of a DGS inside an LMS. This task should be an extension of the work already mentioned above and should be aimed at the support of a blending learning environment of an actual course of “Geometria” for a 1st/2nd year Bachelor course (the Portuguese “Licenciatura”) in Mathematics.

As a second step we want to implement a LMS module which should allow the use of an ATP for the Euclidean geometry and with the capability of axiomatic verification of the soundness of the constructions, inside an LMS (Janičić & Quaresma, 2007). This module should introduce the axiomatic reasoning to the student, this should be an extension of the work done in the 1st stage, mainly in the form of exercises where the difference between an axiomatic theory and its models can be put in evidence.

Then we want to implement a LMS module which should allow the use of an ATP for the Euclidean geometry, inside an LMS (Janičić & Quaresma, 2006; Quaresma & Janičić, 2006b; Quaresma & Janičić, 2007). With this we want to introduce the axiomatic reasoning, exploring geometric proofs done by the ATP i.e. proofs using a geometric reasoning, that can be read and understood by the student.

Finally we want to build an *adaptive learning environment*. A system where all the components mentioned above, and a repository of geometric problems should be integrated (Quaresma & Janičić, 2006b; Quaresma & Janičić, 2007), allowing the student to have a broad experimental platform.

A. Tools

In this section we present a very brief description of the different tools (most of this description were taken from their own Web-pages) that were referred in this text, given also the URLs of their Web-pages.

DGS – Cabri Cabri Geometry is an interactive geometry software for teaching and learning geometry and trigonometry. It was designed with ease-of-use in mind. The program allows the user to animate geometric figures. Relationships between points on a geometric object may easily be demonstrated, which can be useful in the learning process. There are also graphing and display functions which allow exploration of the connections between geometry and algebra. Commercial distribution, MS-Windows + Mac OS, <http://www.cabri.com>.

DGS – Cinderella Interactive geometry and analysis takes place in the realm of euclidean geometry, spherical geometry or hyperbolic geometry. It includes a physics simulation engine and a scripting language. Commercial distribution, Java-enabled platforms, <http://www.cinderella.de/tiki-index.php>.

LMS - Claroline Claroline is an Open Source eLearning and eWorking platform allowing teachers to build effective online courses and to manage learning and collaborative activities on the web. Translated into 35 languages, Claroline has a large worldwide users and developers community. Open source, PHP/MySQL/Web server platforms, <http://www.claroline.net>.

DGS – Eukleides Eukleides is a Euclidean geometry drawing language. It has been designed in order to be close to the traditional language of elementary Euclidean geometry. In many cases, it makes possible to completely avoid the use of Cartesian coordinates. Open source, Linux, Mac OS, MS-Windows, <http://www.eukleides.org>.

DGS/ATP – GCLC GCLC is a tool for visualizing and teaching geometry, and for producing mathematical illustrations. The basic idea behind GCLC is that constructions are formal procedures, rather than drawings. Thus, in GCLC, producing mathematical illustrations is based on “describing figures” rather than of “drawing figures”. WinGCLC is the MS-Windows version of GCLC, with a GUI interface, and providing a range of additional functionalities. Free software, MS-Windows, Linux, Mac OS, <http://poincare.matf.bg.ac.yu/~janicic/gclc>.

DGS – GeoGebra GeoGebra is free and multi-platform dynamic mathematics software for learning and teaching. It has received several educational software awards in Europe and the USA. Free software, Java-enabled platforms, <http://www.geogebra.org/cms>.

DGS - GeometerSketchpad The Geometer’s Sketchpad is a dynamic construction, demonstration, and exploration tool that adds a powerful dimension to the study of mathematics. A user can use this software program to build and investigate mathematical models, objects, figures, diagrams and graphs. Commercial distribution, MS-Windows, Mac OS, <http://www.dynamicgeometry.com>.

RGP - Geometriagon Is a Web system with a large repository of geometric constructions. It contains a java applet that allows the construction of geometric figures with validation of a given geometric property. Web system, free access, <http://www.polarprof.org/geometriagon>.

DGS/ATP - GeoProof GeoProof is an interactive geometry software with proof related features. The project consist in producing an interactive proof software for geometry. GeoProof can communicate with the Coq proof assistant to perform automatic and interactive proofs of geometry theorems. Open source, requires OCML compiler and the Coq proof assistant, <http://home.gna.org/geoproof>.

RGP - GeoThms Is a Web system with a large repository of geometric constructions. It integrates two DGSs and two ATPs, contains a workbench to allow the use of such tools. Web system, free access, <http://hilbert.mat.uc.pt/GeoThms>.

DGS/ATP – JGEX JGEX is a system which combines the authors approach for visually dynamic presentation of proofs, dynamic geometry software and automated geometry theorem prover. Free software, Java-enabled platforms, <http://woody.cs.wichita.edu/>.

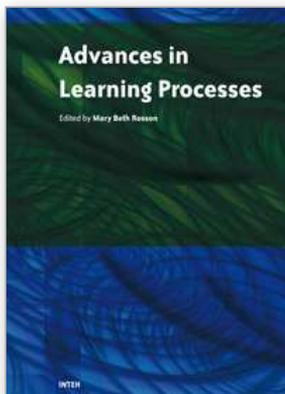
LMS - Moodle Moodle is a Course Management System (CMS), also known as a Learning Management System (LMS) or a Virtual Learning Environment (VLE). It is a Free Web application that educators can use to create effective online learning sites. Open source, PHP/MySQL/Web server platforms, <http://moodle.org>.

LMS - WebCT WebCT is an online proprietary virtual learning environment system that is sold to colleges and other institutions and used in many campuses for e-learning. To their WebCT courses, instructors can add such tools as discussion boards, mail systems and live chat, along with content including documents and web pages. Commercial distribution, MS-Windows, Linux, Solaris, <http://www.blackboard.com>.

B. References

- Buchberger, B., Craciun, A., Jebelean, T., Kovacs, L., Kutsia, T., Nakagawa, K., Piroi, F., Popov, N., Robu, J., Rosenkranz, M. & Windsteiger, W. (2006). Theorema: Towards computer-aided mathematical theory exploration, *Journal of Applied Logic* 4(4): 470–504.
- Chou, S.-C. (1987). *Mechanical Geometry Theorem Proving*, D. Reidel Publishing Company, Dordrecht.
- Chou, S.-C., Gao, X.-S. & Zhang, J.-Z. (1993). Automated production of traditional proofs for constructive geometry theorems, in M. Vardi (ed.), *Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science LICS*, IEEE Computer Society Press, pp. 48–56.
- Chou, S.-C., Gao, X.-S. & Zhang, J.-Z. (1994). *Machine Proofs in Geometry*, World Scientific.
- Chou, S.-C., Gao, X.-S. & Zhang, J.-Z. (1996a). Automated generation of readable proofs with geometric invariants, I. multiple and shortest proof generation, *Journal of Automated Reasoning* 17: 325–347.
- Chou, S.-C., Gao, X.-S. & Zhang, J.-Z. (1996b). Automated generation of readable proofs with geometric invariants, II. theorem proving with full-angles, *Journal of Automated Reasoning* 17: 349–370.
- Chou, S.-C., Gao, X.-S. & Zhang, J.-Z. (1996c). An introduction to geometry expert, *Automated Deduction – CADE13*, Vol. 1104 of *Lecture Notes in Computer Science*, Springer, pp. 235–239.
- Coelho, H. & Pereira, L. M. (1986). Automated reasoning in geometry theorem proving with prolog, *Journal of Automated Reasoning* 2(4): 329–390.
- Janičić, P. (2006). GCLC — a tool for constructive euclidean geometry and more than that, *Lecture Notes in Computer Science* 4151: 58–73.
- Janičić, P. & Quaresma, P. (2006). System Description: GCLCprover + GeoThms, in U. Furbach & N. Shankar (eds), *Automated Reasoning*, Vol. 4130 of *LNAI*, Springer, pp. 145–150.
- Janičić, P. & Quaresma, P. (2007). Automatic verification of regular constructions in dynamic geometry systems, *Automated Deduction in Geometry*, number 4869 in *LNAI*, Springer, Berlin / Heidelberg, pp. 39–51. 6th International Workshop, ADG 2006, Pontevedra, Spain, August 31-September 2, 2006. Revised Papers.
- Matsuda, N. & Vanlehn, K. (2004). Gramy: A geometry theorem prover capable of construction, *Journal of Automated Reasoning* 32: 3–33.
- Narboux, J. (2004). A decision procedure for geometry in Coq, *Lecture Notes in Computer Science* 3223: 225–240.
- Quaresma, P. & Janičić, P. (2006a). Framework for constructive geometry (based on the area method), *Technical Report 2006/001*, Centre for Informatics and Systems of the University of Coimbra.
- Quaresma, P. & Janičić, P. (2006b). Integrating dynamic geometry software, deduction systems, and theorem repositories, in J. M. Borwein & W. M. Farmer (eds), *Mathematical Knowledge Management*, Vol. 4108 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 280–294.
- Quaresma, P. & Janičić, P. (2007). GeoThms - a Web system for Euclidean constructive geometry, in S. Autexier & C. Benz Müller (eds), *Proceedings of the 7th Workshop on User Interfaces for Theorem Provers (UITP 2006)*, Vol. 174 of *Electronic Notes in Theoretical Computer Science*, Elsevier B.V., pp. 35–48.

- Quaresma, P., Janičić, P., T. J., V.-Jančić, M. & Tošić, D. (2008). *Communicating Mathematics in The Digital Era*, A. K. Peters, Ltd., chapter XML-Bases Format for Descriptions of Geometric Constructions and Proofs, pp. 183–197.
- Robu, J. (2002). *Geometry Theorem Proving in the Frame of the Theorema Project.*, PhD thesis, Johannes Kepler Universität, Linz.
- Santos, V. & Quaresma, P. (2008). eLearning course for Euclidean Geometry, *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies, July 1st- July 5th, 2008, Santander, Cantabria, Spain*, pp. 387–388.
- Team, T. C. D. (2007). *The Coq Proof Assistant, Reference Manual, Version 8.1*, Projet Coq, INRIA, Lyon, France.
- Wisher, R. (2009). *SCORM 2004*, 4th edn, Advanced Distributed Learning.
- Zhang, J.-Z., Chou, S.-C. & Gao, X.-S. (1995). Automated production of traditional proofs for theorems in euclidean geometry I. the Hilbert intersection point theorems., *Annals of Mathematics and Artificial Intelligence* **13**: 109–137.



Advances in Learning Processes

Edited by Mary Beth Rosson

ISBN 978-953-7619-56-5

Hard cover, 284 pages

Publisher InTech

Published online 01, January, 2010

Published in print edition January, 2010

Readers will find several papers that address high-level issues in the use of technology in education, for example architecture and design frameworks for building online education materials or tools. Several other chapters report novel approaches to intelligent tutors or adaptive systems in educational settings. A number of chapters consider many roles for social computing in education, from simple computer-mediated communication support to more extensive community-building frameworks and tools. Finally, several chapters report state-of-the-art results in tools that can be used to assist educators in critical tasks such as content presentation and grading.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Vanda Santos and Pedro Quaresma (2010). Adaptative Learning Environment for Geometry, *Advances in Learning Processes*, Mary Beth Rosson (Ed.), ISBN: 978-953-7619-56-5, InTech, Available from: <http://www.intechopen.com/books/advances-in-learning-processes/adaptative-learning-environment-for-geometry>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.