Int J Comput Math Learning (2007) 12:57–79
DOI 10.1007/s10758-007-9111-5

AgentGeom: a multiagent system for pedagogical support in geometric proof problems

Pedro Cobo · Josep M. Fortuny · Eloi Puertas · Philippe R. Richard

Published online: 23 March 2007

Abstract This paper aims, first, to describe the fundamental characteristics and workings of the AgentGeom artificial tutorial system, which is designed to help students develop knowledge and skills related to problem solving, mathematical proof in geometry, and the use of mathematical language. Following this, we indicate the manner in which a secondary school student can appropriate these abilities through interactions with the system. Our system uses strategic messages of the agent tutor in an argumentative process that collaborates with a student in the construction of a proof.

Keywords AgentGeom · Multiagent system · Pedagogical support · Geometric proof problem · Appropriation · Interactions

P. Cobo
Departament de Matemàtiques, Institut d'Ensenyament Secundari Pius Font i Quer de Manresa, Manresa, Spain
e-mail: pcobo@xtec.net

J. M. Fortuny (✉)
Departament de Didàctica de la Matemàtica i de les Ciències Experimentals, Universitat Autònoma de Barcelona, Bellaterra, Spain
e-mail: jfortuny@uoc.edu

E. Puertas
Institut d'Investigació en Intel·ligència Artificial, Consell Superior d'Investigacions Científiques, Espanya, Spain
e-mail: Eloi@iiia.csic.es

P. R. Richard
Département de didactique, Université de Montréal, Montreal, Canada
e-mail: Philippe.r.richard@umontreal.ca

1 Introduction

Argumentation has a privileged position in the acquisition of knowledge in a mathematics class and social debate and discursive logic are at the heart of mathematical discovery. The constructivist perspective requires the reproduction of the characteristics constitutive of mathematical work: *''We know that the only means of making mathematics, is to seek and solve certain specific problems and, on this subject, to put forth new questions''* (Brousseau 1997). Since personalized attention and the encouragement of debates in class are didactically expensive (the time needed to devote to it, the teaching means mobilized simultaneously to ensure the solving problem process and the respect of the internal logic of the situation for each student), it can be understood why teaching practice resorts to it only occasionally.

Our paper aims to show how a secondary school student can appropriate geometrical abilities through interactions with an agent tutor in an artificial tutoring system (hereafter called ''AgentGeom''). The kinds of interactions that interest us are discursive in nature (the exchange of messages expressed in natural language or a symbolic language) or graphic (the exchange of messages expressed in the register of geometrical figures) between students and their artificial tutor. The application of this model of interactions based on research into teaching mathematics is programmed into AgentGeom, with the aim of giving students cognitive and metacognitive support to help them in the development of their problem solving ability and in mathematical reasoning.

This article begins with a general description of the ways in which teachers can prepare a work session for their students in AgentGeom, and how students can use the AgentGeom student interface to carry out the tasks set for them by the teacher. We then describe some technical details of the tutorial system architecture (Sect. 2). In Sect. 3 we show the interactions of AgentGeom with one of the students that participated in the experiments we carried out, in order to test the system. This general view of the operation and potentiality of AgentGeom sets the stage for the presentation of our theoretical framework, which is discussed in Sect. 4. We also analyse the cognitive benefits for students arising from their interactions with the system. We conclude by summarizing the characteristics of AgentGeom and its implementation, its limitations and possible future development.

2 How to use AgentGeom

The descriptions of the teacher and student interfaces will give the reader an idea of the system's communication possibilities: with the teacher, providing him/her with various ways of preparing and assigning tasks to students; with the students, helping them to produce graphic and deductive actions, and showing them help messages guiding their problem-solving processes.

2.1 How teachers can prepare a work session for students

When teachers enter the system to prepare a work session for students, they have two options: to use the problems that the AgentGeom tutorial system has already implemented, or to create their own problems.

AgentGeom places a range of geometric problems at the teacher's disposal—problems that compare areas—which have been extensively studied and tested with students in various contexts (Cobo 1998, 2004). These problems are classified according to whether they can be solved using one, two or more strategies. The teacher can prepare a problem itinerary—simply by selecting them—and assign them to a student (or to several) taking his/her cognitive characteristics into account. To do so, the teacher only has to open the ''problems assignment'' interface, to which only the teacher has access, select the problems and click on the student's name (see Fig. 1).

The AgentGeom tutorial system also offers teachers the possibility of implementing their own problems. To use this option, the teachers first need to identify the various strategies that resolve the problem under consideration, so that they can make a reasonable forecast of students' actions. Secondly, the teacher must prepare a document with pedagogical messages that provide differing levels of information, and are grouped according to the phases of the solving processes which are being carried out—familiarization, planning, execution, etc. Finally, teachers must prepare the figures that they want the system to show students.

2.2 How students can use AgentGeom to carry out the tasks proposed by the teacher

The student's interface (Fig. 2) provides students with all the necessary tools for resolving a geometrical problem. The left hand side acts as a graphic area in which the student can build a geometrical figure by using buttons on the left to place points or segments, to build parallel or perpendicular straight lines, to define the intersection of two objects, etc.

The right hand side contains the deduction editor, which the student can use to test, compare measurements of constructed graphical objects or verify the relations of parallelism or perpendicularity between straight lines. The combined use of the deduction editor and the graphic area allows the pupil to construct inferences (see Sect. 4.2 about discourse).

Furthermore, using their interface, students can at any time see the statement of the problem (see bottom left of Fig. 2), ask for a help message, or load any graphic instructions that the teacher has included to help students. At any time, students may also open their record window in which both the teacher and the student can see complete information on

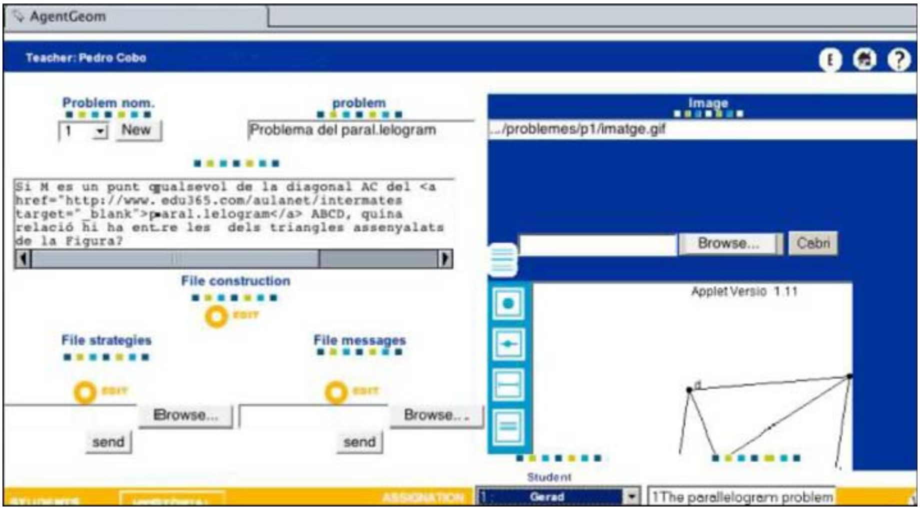


Fig. 1 Part of the teacher interface for implementing new problems

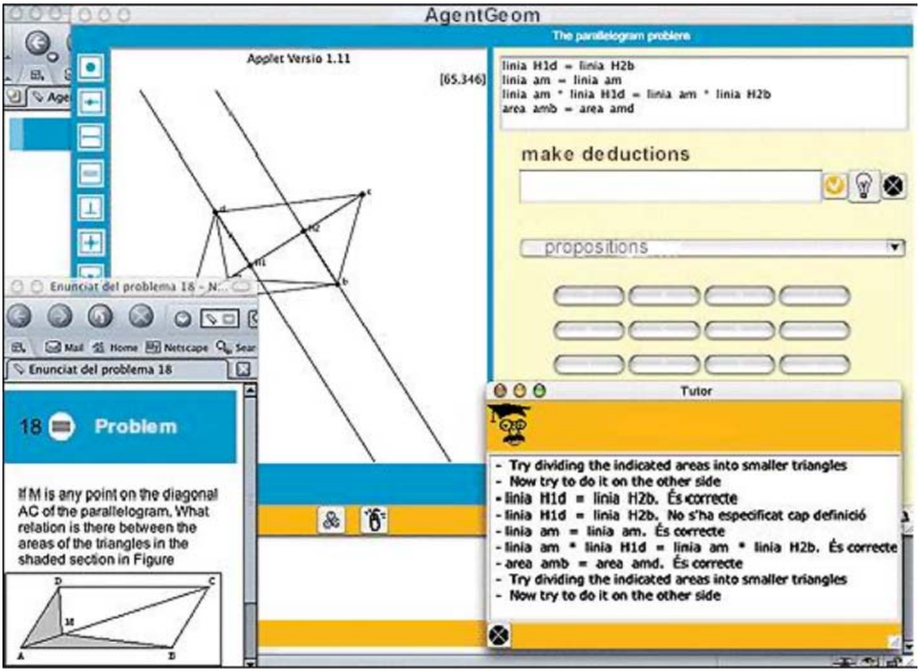


Fig. 2 Part of the student interface

all the student's actions—graphic and deductive—and their validations, the messages sent by the tutoring agent, whether or not they were requested by the student, and the day and time on which all the actions were carried out. The contents of this window will be a source of very important data for analysis of students' actions in solving the problems proposed by both the teacher and researcher.

2.3 AgentGeom architecture and basic definitions

AgentGeom combines two basic attributes: it is open and it allows attention to diversity in the sense that the teacher can choose between various types of geometrical problems that are adaptable to the specific needs of each student. We created portable software developed on a web architecture; the system is on a server and it is available to a group of students by means of any web browser. The teachers and students can interact easily with the system and start with any normal computer connected to a network. It can work with inexpensive computers of the sort found in schools. AgentGeom also incorporates a virtual debate forum that facilitates interaction between students and with the teacher.

Furthermore, we should mention that AgentGeom is an *artificial tutoring multi-agent system*. It is like the old intelligent tutor systems, but benefits from the open and modular capacity that multi-agent systems provide. The elements that comprise the AgentGeom system are as follows:

- *Interfaces:* These include all the available tools that users (students, teachers) can use to interact with the other agents.
- *Agent Mediator:* receives all inputs from the interfaces and processes them.

- *Agent Tutor:* whose main goal is to give advice to the student.
- *Database:* where Agents enter all the data collected from the users.

Figure 3 shows how the different elements are connected to each other.

2.3.1 Agent mediator

The agent mediator receives all inputs from the teacher and student interfaces. Firstly, problems collected from the teacher interface are stored in the database, so the next time a student enters the system, he or she can choose one of these new problems and start to solve it. Lastly, the agent mediator receives all the actions that a student has made on a selected problem. The architecture of the agent mediator benefits from previous work by Luengo (1999).

We define a geometrical action as any action that a student takes in the graphical area, and a deductive action as those actions made using the deduction editor. A geometrical action is an elemental geometrical construction, which we will call an EGC, and some associated parameters. Some EGCs have special parameters, for example, to create a point you may choose its position. Other parameters are related to other elements already existing in the figure; for example, if you click on the mid-point in a line, you may choose an additional segment. Deductive actions have a special format that is applied by the deduction editor.

The agent mediator executes different tasks depending on the kind of action taken, geometrical or deductive. Geometrical actions let the agent mediator keep an internal representation of the figure that the student is constructing. We call this representation the geometrical model. The geometrical model must have all the knowledge from classical metric geometry that the student needs to solve any kind of geometrical problem. We therefore organized all this knowledge in a frame representation (Winston 1992), where each geometric entity is a different class. Each class has several descriptors or properties,

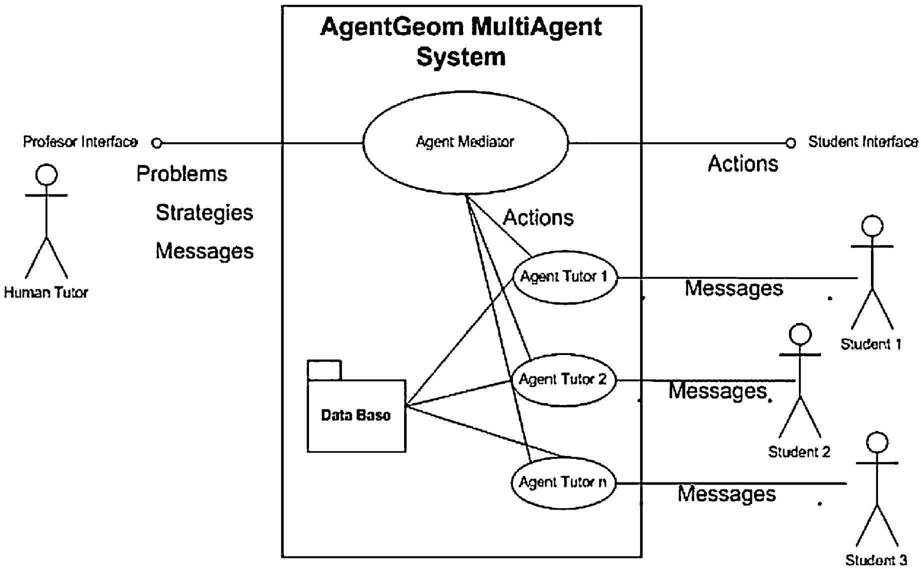


Fig. 3 AgentGeom architecture

such as: name, area, number of vertex, degrees, length, etc. Afterwards, the agent mediator only has to create new, suitable objects and fill in all properties for that new entity. Some properties will be set with the parameters specified in the geometrical action; others will be filled in automatically.

To conclude, the agent mediator takes the following steps when performing a geometrical action: add the EGC specified in the action to the geometric model; calculate all new entities derived from that action and if the process ends correctly, show the resulting figure; otherwise, show an error.

Deductive actions are those used by students to check properties inside the geometrical figure he or she is building. These actions must have a well-formed syntax and the agent mediator is the one who checks if they are correct. We use a syntax similar to one used in the automated geometry problem solving area (Matsuda and Okamoto 1998). Following that syntax, each action consists of three parts—the left side, operator and right side. The left side is an instance from the model, including completed elements such as points, lines, segments, circles, angles and polygons. An operator can be one of the following—equal, less than, more than, parallel to, perpendicular to, the area of a polygon or the degrees of an angle. The right side can be either an instance or a specific value (see Fig. 2).

The agent mediator checks if any element expressed in the action exists within the model, and if the operator can be applied to both sides of the deduction. A student can discover or deduce a value or relation between different elements. The agent mediator then only has to say whether the deduction is correct or not. To do so, it only has to look at specific properties of the elements cited within the geometrical model, and then apply the selected operator to them. The agent mediator only assesses individual actions. It is the agent tutor's job to decide if the student's actions can lead him or her to the solution or not. When the agent mediator receives a deductive action it acts as follows: check deduction syntax, check deduction validity using the geometric model and show the deduction result.

2.3.2 Agent tutor

The agent tutor's goal is to help the student to resolve the given problem. It communicates with the student by using messages. So, a dialogue is maintained between the agent tutor and the student. The student carries out actions and the agent tutor gives advice. In the following sections, we will analyse the didactical foundations of this kind of dialogue.

The agent tutor needs to know when it should send a message and what the message must say. To do so, the agent tutor needs to first receive the mathematical knowledge involved in solving the proposed problem, which an expert may identify to anticipate possible problems in the student's work. The teacher enters the initial knowledge for this problem into the database using his own interface, as mentioned above. Specifically, the agent tutor receives all strategies and messages for the problem. As mentioned above, the strategies specify several paths for solving the problem.

The agent tutor keeps all strategies in a ''forest'' structure. This is a set of tree structures, in which each strategy is a different branch of a tree (see Fig. 6). The nodes of these trees are actions that include an inference. Actions contained in this tree are the only ones recognised by AgentGeom when validating the argumentation proposed by the student.

A special action is the solution or solutions to a problem. If a student wants to give the solution to the problem, then the agent checks the solution and, if is correct, evaluates how well-reasoned the solution is. To do this, the agent tutor summarizes how many recognized actions the student has carried out in each strategy, and then he weighs each strategy with an ad-hoc heuristic to obtain a final mark. If that mark is higher than a specific threshold, the agent tutor comes to the conclusion that the student has correctly solved the problem with some theoretical support.

Furthermore, the agent tutor keeps different lists of messages. There is a different list for each strategy and a special list called change strategy messages. Each message is associated with one node defined on the strategy tree. It is possible to assign more than one message in a single node, then it can be decided if the messages for that node will be displayed at random or in a specific order.

The agent tutor therefore has all the information about messages and strategies, and only needs a mechanism to know when and what message should be displayed. This mechanism starts when the agent mediator, who tracks all actions performed by the student, transfers them to the agent tutor. The agent tutor checks every action inside its strategy trees. The agent tutor can thus locate where the student is within the resolution path. When the agent tutor realizes that a student is no longer following a valid path (e.g., when the student's last actions were not recognized, or a student does not send any more actions), the agent tutor sends a message to the student. The message sent is the one attached to the last node in the current strategy followed by the student up until then, i.e., the message associated with the last recognized action. If the message selected was previously shown, then the agent tutor selects another message for this node if there are any available. In nodes marked as random, the agent tutor chooses a message that has not yet been selected. Otherwise, he uses the next message on the list for that node. When all messages for a single node have been displayed, the agent tutor shows a different message from the change strategy list. The purpose of the messages on this list is to force the student to change the strategy, i.e., the message displayed gives some advice that allows the student to execute a recognized action from another strategy (see Fig. 4).

3 Test of a Case Study though Interaction with the AgentGeom

This case study describes how students can learn from AgentGeom by participating in joint geometrical proof problem solving. They appropriated aspects of mathematical practices such as conceiving what a mathematical proof is and how the sentences are written and validated, which are inferred from the figural representation of the statement of the problem. The students also exteriorise the need to articulate a sequence of sentences (some of which stem one from another) that was previously validated or makes use of an already established geometric property.

The study focuses on two aspects of appropriation: what learners appropriate and how learners actively transform what they appropriate. We describe in detail how students appropriate ways of seeing a proof, react to the messages of the agent tutor, change focus during the process of solving, and know when the production of a series of sentences, which the system shows, is good enough to be considered as a solution. AgentGeom habituates the students to the concept of the need to produce a proof as a series of sentences that have to be validated.

3.1 Tasks, participants and setting

In the following paragraphs, we summarize the characteristics of one of the problems that we implemented and of a specific student who is participating in the experiment, as well as the manner in which data was collected. This will allow us to identify and analyse the student's interactions with AgentGeom and the cognitive benefits stemming from these interactions.

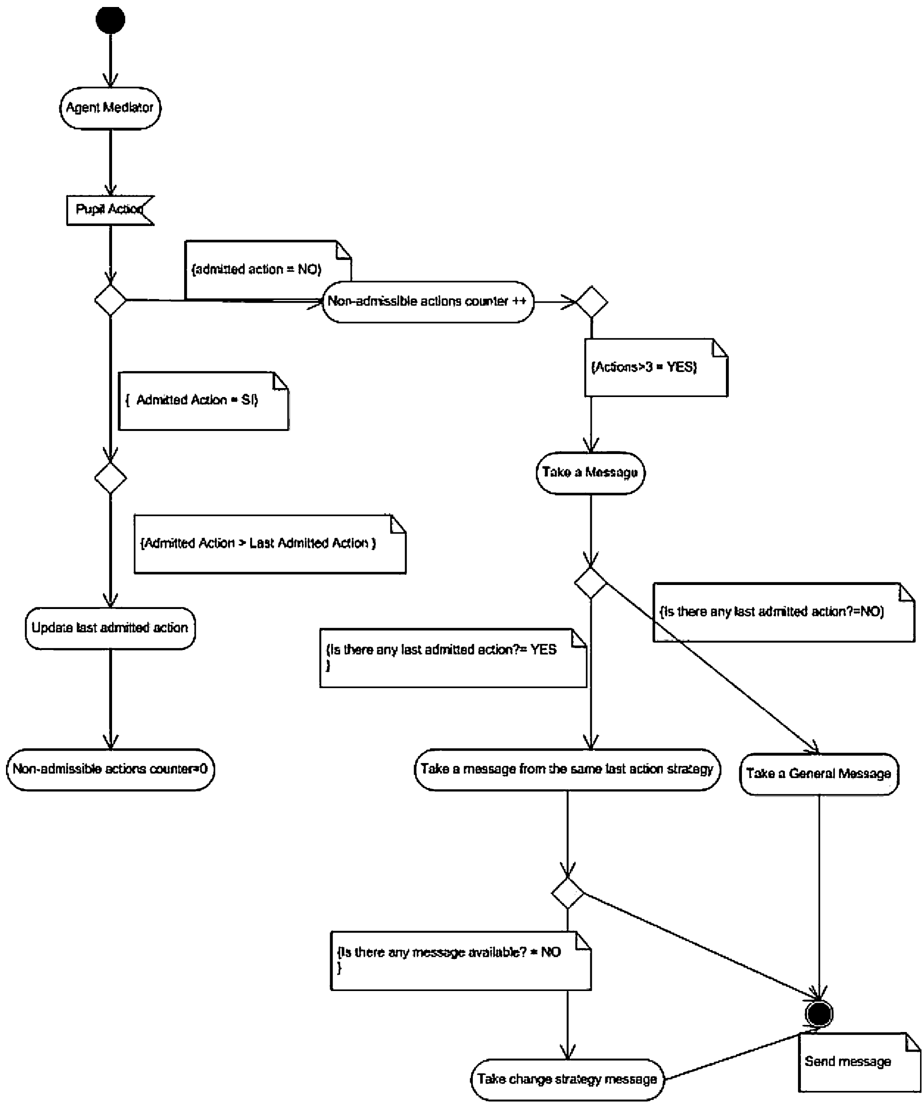


Fig. 4 Agent tutor message selection algorithm

We selected a problem of calculation and comparison of areas of flat figures that we call the parallelogram problem (Fig. 5), since this solution can be arrived at in different ways and because there is the possibility of solving it using a combination of graphic and deductive components.

Based on the construction of its solution tree (Fig. 6), we analysed the characteristics of this problem in depth, and identified the conceptual and procedural concepts involved in solving it (Cobo 1998), as well as all the possible graphic and deductive actions that a student might perform to arrive at a solution to the problem. In short, we can say that there are five ways of reaching a solution and they are those that we implemented in

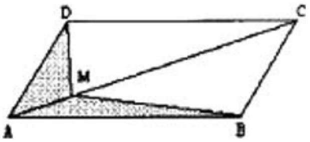


Fig. 5 Parallelogram problem statement

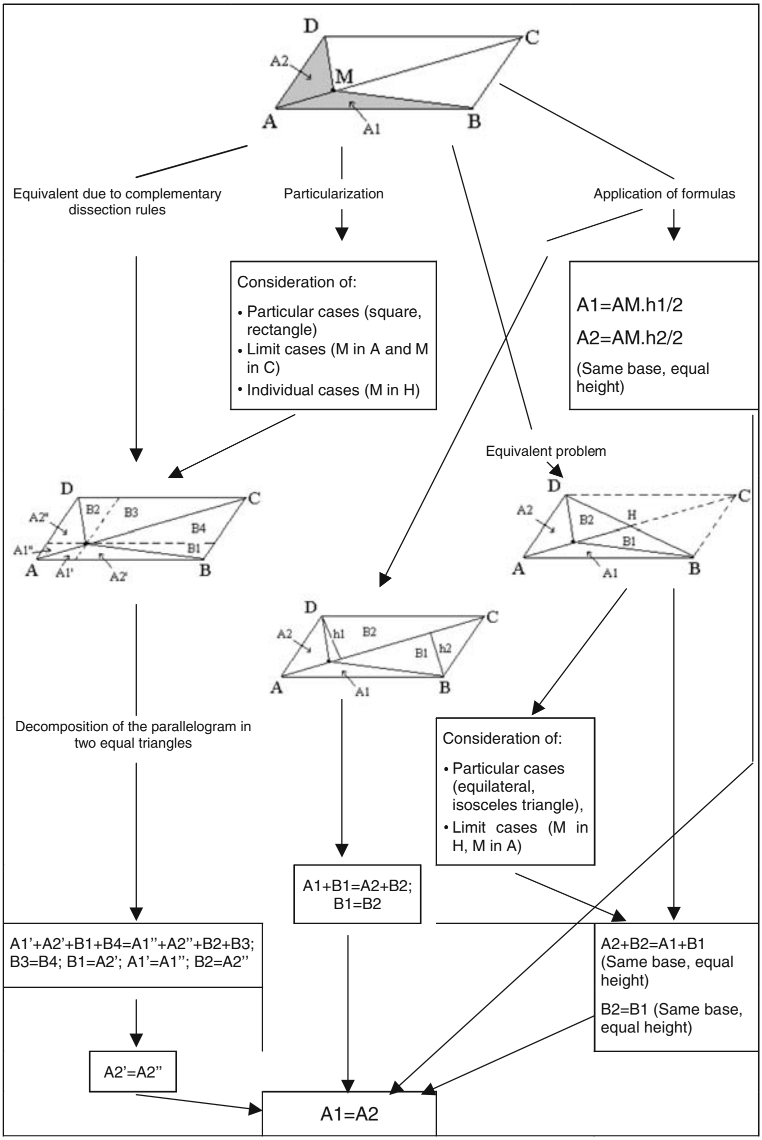


Fig. 6 Solution tree of the parallelogram problem

AgentGeom. We call them: ‘‘equivalent due to complementary dissection rules’’, ‘‘particularization’’, ‘‘equivalent problem’’, and two ways based on ‘‘the application of formulas’’ of the area of a triangle.

Several students participated in the experimentation with AgentGeom. The data that we analyse in this article is taken from a 16-year-old student named Gerard, who is studying in his first year of secondary school in Manresa, Catalonia.

Prior to his interaction with AgentGeom, Gerard did not receive any specific training on problem solving. On the contrary, the teaching in this subject has always been focused on solving applied exercises or problems, the mathematical contents of which the teacher had just finished explaining. The work method that was followed in Gerard’s class combines expositional phases by the teacher and a wide range of activities that the students carry out in groups of two or three—they rarely work individually—and in which they have a large degree of freedom to comment on the activities. Gerard and his classmates frequently use the classroom’s computers, always in small groups, carrying out structured and guided activities, and always associated with the mathematical content that is being taught. They therefore have no individual experience or a high degree of autonomy in dealing with more open problems not directly related to the subjects that the class is studying.

The mathematical content of the problem that we propose—comparison of areas—was dealt with in courses prior to the one Gerard is currently taking. Gerard therefore has procedural knowledge related to the application of formulas for calculating the area of flat figures, and sufficient knowledge of the concepts associated with the geometric constructions of the graphic area of AgentGeom (points, straight lines, segments, parallels, perpendiculars, etc.) and those that are used in writing the deductive sentences (area, relationships of equality and inequality, length of segments, etc.). Nevertheless, Gerard has never had any specific teaching aimed at the comprehension of proof in mathematics and, therefore, does not have any argumentative or deductive abilities within a recognized model, which is precisely one of the cognitive benefits that can be obtained from interactions with AgentGeom.

Under these conditions, each one of the students, participating in the experiment, (and Gerard in particular) is given a detailed explanation of how AgentGeom works, and they are asked to solve the parallelogram problem with the help of the agent tutor. The data from this solution are recorded in the student’s case history. Based on this case history, and on the transcription of the recordings of everything that took place on the screen using ScreenFlash (2002) software, we have obtained the written protocols that we use in the characterization of the interactions models.

3.2 Analysis of the student’s cognitive progress resulting from interactions with AgentGeom

This section summarizes the bases of the microanalysis. It provides the theoretical framework of the interactions between the student, the agent tutor and the work area in order to examine in detail the development of skills operating during the interactions and how the student was active in appropriating proof abilities. To be more precise:

a) We analyse the joint use of the graphic area and the deductive area, i.e., the different phases in which the student uses either one, the other, or both of them, as well as the interventions by AgentGeom and the evolution of his knowledge in each phase.
b) We study the proactive and reactive nature of the actions of the student and their influence on the solving process. The action is of the *proactive type* if the student

proceeds on his/her own initiative, without waiting for information from the agent tutor. If the system comes to invalidate the well-formed character of an element of a figure or a deduction, the action of the student is still proactive if he or she engages in the rectification of the form, but the engagement does not respond to a message from the agent tutor. The action is of a *reactive type* when the student answers a message from the agent tutor and this action follows what is suggested by the message.

c) We analyse the influence that the messages of the agent tutor has on the student. For example, we examine whether the student actually followed the suggestion of the agent tutor and this suggestion was beneficial to him or, if on the contrary, the student was unaware of the message and continued on his/her own initiative. This influence is examined based on three levels of messages from the agent tutor. Level 0 contains general type suggestions, i.e., suggestions that do not imply mathematical contents or procedures in the solving process (''Organise the information that you have'', ''Try to understand the conditions of the problem'', ''Review the process that you have followed'', etc.). The messages of level 1 only convey the name of the implied mathematical contents or procedures (''Remember that this is a parallelogram'', ''Think about the point of the diagonal that should be m according to the problem statement'', ''Try to explain what the diagonal of a parallelogram is'', etc.). Level 2 provides more specific information on these contents or procedures (''A parallelogram has parallel sides two by two'', ''The diagonals of a parallelogram can be divided at their midpoints'', ''As bases for triangles, you can assume that they compare the common side am'', etc.).

To analyse the problem solving process, we have divided his transcript into social episodes, which are periods of time in which the student completes a phase of the process followed, as defined by Schoenfeld (1985).

We have identified three social episodes in Gerard's problem solving process of the parallelogram. We have named them according to the purpose of the student's actions. If the student's purpose is clear, we will be looking at an episode of analysis. If, within the same episode, there is a combination of actions that do not seem to have a clear purpose with others that do have one, we shall call it exploration/analysis. Finally, if the student's aim is to prove the conjecture that he or she proposes, we shall call it an episode of implementation or justification.

3.2.1 An episode of exploration/analysis

When starting the problem solving process, after reading the explanation of the problem, Gerard tries to use the only abilities he has for solving problems that involve equivalencies of triangles. His abilities include the use of procedures related to identifying their heights so as to apply formulas that allow him to calculate their areas. With this intention, Gerard begins by drawing two lines that are perpendicular to the *dc* and *ba* sides, going through *b* and *d*, respectively (actions 3 and 4). But he rapidly erases them when he discovers that they do not match the heights of any triangle (Fig. 7a) (Table 1).

Drawing the perpendicular line to segment *ac*going through point *d* (Fig. 7b), and the *pd* height of the triangle *amd* (actions 8, 9 and 10), presents Gerard with one of the problem solving strategies, which we have named: ''application of formulas'' (Fig. 6), which requires drawing the heights of the triangles as initial actions.

Gerard is therefore on the right path. He started the process by using the AgentGeom graph area. The agent tutor, who identified the thought process that Gerard seems to be

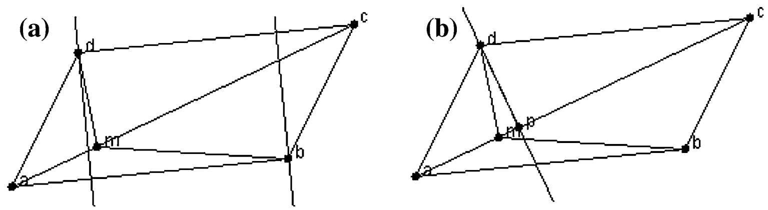


Fig. 7 Auxiliary lines drawn by Gerard

Table 1 How Gerard explored the parallelogram problem solving process

2	Gerard	Downloads the figure
3	Gerard	Creates a line perpendicular to segment dc going through point b named: l
4	Gerard	Creates a line perpendicular to segment ba going through point d named: j
5	Gerard	Erases line j
6	Gerard	Erases line l
7	Gerard	Creates a line perpendicular to segment ac going through point d named: s (cancels)
8	Gerard	Creates a line perpendicular to segment ac going through point d named: s
9	Gerard	Creates an intersection from segment s to segment ac named: p
10	Gerard	Creates a line going through point p and going through point d named: pd
11	Gerard	Erases line s
12	Gerard	Looks for messages from the tutor's window; there are none

trying to follow remains silent, so that the student will continue taking recognized steps using strategies that obtain the solution to the problem, and to maintain the characteristic of non-intervention by the agent tutor.

When a teacher and student interact, they each know how to interpret the other's non-verbal forms of communication especially if they have been working together for a long time. This time the agent tutor has remained silent and Gerard did not now how to interpret that, despite an explanation of how AgentGeom functions being given to the students before starting the experience. It may be possible that the students and AgentGeom need a longer period of mutual adaptation for improved understanding of the communication processes.

After looking at the tutor's window and not seeing any message (action 12), Gerard begins to abandon the strategy that he seemed to be following. If he had made a request for help, the agent tutor would have encouraged the student to ‘‘identify the heights of the triangles’’ and to ‘‘compare their sides’’. We highlight this hypothetical action by Gerard to underline the potential of AgentGeom. Had Gerard made this request and followed the tutor's suggestion, he might of learned more about how triangles can be compared by comparing their elements.

But why didn't Gerard request a message from the agent tutor? And even though he didn't request one, why did he abandon the strategy that he had started?

As far as the first question is concerned, although the professor insisted that the students ask for help from the agent tutor at key points before beginning the experience, we observed a desire to solve the problems with the least help possible in all of the students' actions. This is a possible justification for Gerard's performance.

There could be many answers as to why Gerard abandoned his strategy, and possibly they should be sought in the cognitive characteristics of students of this age. For example, students are not accustomed to work with heights that ‘‘are not within the base of the triangle’’, as Cobo (1998) demonstrates, or not knowing any numerical data and, therefore, to their way of thinking, being unable to calculate the area of any triangle. This makes many students desist from applying the formula for the area of the triangle. Because in many cases, this is their only way of approaching the solution of the problem, they may then reach an impasse and be unable to proceed. They do not understand that all that the problem requires is that the length of the opposing sides of the parallelogram are equal; the exact length of those sides is irrelevant for the proof.

In short, at the start of the solving process, Gerard interacted with the graphic part of AgentGeom in a proactive way. He drew on his knowledge of mathematics that has to do with drawing perpendicular lines, and on the recognition of one of the heights of the triangles involved in the solution. Gerard did not request any message nor did he produce any productive action, therefore, the agent tutor did not explicitly intervene in this episode.

3.2.2 Episode of analysis

More than six minutes have gone by since Gerard abandoned the strategy of application of formulas that the drawing of the heights of the triangles requires. In this time, he has erased all of the lines that he found and he has moved about the AgentGeom screen without doing anything. He is lost. He does not know how to continue on. The opening of the procedures window and the choosing of the word ‘‘median’’ (action 23) mark the beginning of a new episode, of the *analysis* type, and it is one that will lead Gerard to propose a conjecture for the problem of the parallelogram.

Gerard’s performances in this episode are marked by three key actions. We have highlighted the first:

23. Gerard: [Opens the procedures window and chooses the median line]

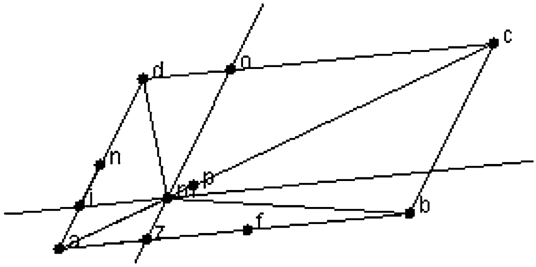
This action is the beginning of a series of graphic actions having a specific objective: describing the perpendicular bisector of side ‘‘*da*’’. The agent tutor does not recognize any of these actions. They do not make up part of any solution strategy. The agent tutor therefore begins to react by sending Gerard the first message (action 28), which is the second key action of this episode.

28. Agent Tutor: Try to compare the sides of the triangles.

The tutor takes the last recognized action –drawing of the *pd*height (action 10)- and sends a level 1 message so that Gerard will continue with that line of solution. The message is, however, very general and gives little information. It is a level 1 message, and, it is therefore a suggestion that only contains the name of the mathematical content and procedure involved. The hidden intention is that the student, by comparing sides, figures out that the *am* side is common to both triangles, and reaches a solution for the problem by comparing the heights of the triangles, using this side as the base.

Gerard is no longer in the same situation as when he started this strategy, because he has erased all the lines. Despite this, he reacts rapidly and uses the deductive area for the first time to compare the da and ba sides.

Fig. 8 Auxiliary lines drawn by Gerard



29. Gerard: line da 2 = line ba*

30. Agent Tutor line da 2 = line ba false*

So far, Gerard has responded to the stimuli and actions of AgentGeom with progressive actions even though, in some cases, they were not recognized by the agent tutor (drawing the median line), and in others the deductions entered were false.

From this point onwards, the communication between the agent tutor and Gerard begins to flow more smoothly. When Gerard starts to use the deductive area, he dares to request a message for the first time (action 34). We consider this to be the third key point in this episode, because it leads Gerard to establish a conjecture. The agent tutor, considering that he did not make the most of the previous message, answers him with a level 2 message that implies a change of strategy (action 35). The first part of this message contains general information: ''*Could you think of some way to break the parallelogram down into triangles?''* which is later specified with more precise information: ''*try to draw parallel lines that go through point m*''.

In action 38, Gerard reacts to the agent tutor's message by combining graphic and deductive actions. The graphic actions are a direct response to the content of the message, and those that are deductive are the result of the inspection of the figure after having drawn the lines through m parallel to the da and ba sides (Fig. 8). Gerard begins to make the most of this discursive-graphic combination that gives meaning to the argumentative process (Richard 2004b), and which we consider to be fundamental in the construction of the AgentGeom's architecture (Table 2).

The combination of these graphic and deductive actions, which are clearly progressive in nature, culminates with Gerard's conjecture (actions 40 and 42) that: the areas of triangles *amb* and *amd* are equal. The agent tutor has allowed him to not only establish this conjecture, but has also validated it, as well as validating the conjecture as the final result of the problem (action 43).

In student-AgentGeom communication, it is important to stress the difference between the functions of the deductive action verification button (see validation icon in Fig. 2), which students use when they want to check whether the sentence input is valid or not, and the problem solving verification button (see the light bulb icon in Fig. 2), which as well as validating the last sentence input, shows students whether the system accepts the solving process produced by the student as a valid solution. The system's message will state whether the solution has included the correct result of the problem (even if this result is not the last sentence input) and also has at least 30% of recognized actions (correct actions which are in some of the strategies we have implemented). We have decided to use this percentage, which can easily be modified by the teacher, after analysing various solving processes developed by students while interacting with AgentGeom (Cobo 2004).

Table 2 Part of the episode of analysis/implementation in which Gerard makes a conjecture for the result of the parallelogram problem

34	*Gerard*	*[Requests a message from the agent tutor]*
35	*Agent tutor*	*''Could you think of some way to break the parallelogram down into triangles? For example, by drawing parallel lines that go through point M''.*
36	*Gerard*	*[Tries to erase point n]*
37	*Agent mediator*	*''This point makes up part of one or more lines''*
38	*Gerard*	*Creates a line parallel to segment da going through point m, named: A*
39	*Gerard*	*Creates a line parallel to segment ba going through point m, named B*
40	*Gerard*	*Deduction: area adm = area amb*
41	*Agent mediator*	*Area adm = area amb: true*
42	*Gerard*	*Final deduction: area adm = area amb*
43	*Agents mediator and tutor*	*Area adm = area amb: true*
		Message: I KNOW THAT IT IS TRUE, BUT I DO NOT KNOW WHY. PERHAPS YOU CAN DEDUCE IT STARTING FROM THE ELEMENTS OF YOUR FIGURE

From this point on, Gerard knows that the agent tutor is making some graphic and deductive tools available to him, which may help him to find the solution to a problem, but they require an argumentative process from him, which up until now he has not been accustomed to using.

As a summary of the characteristics of the interaction in this episode, we can highlight on one hand, the evolution in Gerard's behaviour, which went from using just the graphic area to the combined use of the graphic and deductive areas, and on the other hand, Gerard's capacity to react to any stimulus given by AgentGeom, from the observation of the word ''perpendicular bisector'', to the proactive message concerning the comparison of the sides, or the reactive message concerning the drawing of parallel lines. In each case, he follows the agent tutor's instructions until he establishes a conjecture. Furthermore, the obligation that the technological system imposes makes Gerard aware of the need to prove the conjecture that he has established; in other words, we have what is known as ''instrumental genesis'', as regards artefact or tool. In our case AgentGeom only becomes a meaningful instrument after a process of instrumental genesis. During instrumental genesis, the student builds up mental schemes. In these mental schemes technical and conceptual components are interwoven (Rabardel 2001). Gerard therefore has the opportunity to generate an argumentative process using a sequence of sentences that the agent mediator must validate. Gerard and the agent tutor thus contribute jointly to giving meaning to a process of proof.

3.2.3 Episode of implementation (justification)

When the agent tutor informs Gerard about the lack of argumentative support for his conjecture, his window remains open, and Gerard looks at and considers a prior message from the agent tutor on the comparison of the sides of the abm and amd triangles (action 28). This proactive message from the agent tutor marks the beginning and the evolution of the argumentative process that Gerard will try to follow throughout this episode.

If we ignore some grammatical errors in Gerard's deductive sentences and the corrective messages by the AgentGeom, it could be said that this episode is once again characterized by the combination of graphic and deductive actions. The difference compared to the previous episode is that Gerard often writes the deductive sentence before having defined the objects involved in it, which he must create afterwards so that the agent mediator understands and validates his deductions. This makes the deductive activity, logically backed up by the graphic constructions, the focus of Gerard's behaviour in this episode.

In the episode of implementation, Gerard makes graphic actions that involve the identification of the points of intersection of lines *A* and *B* (actions 38 and 39) with sides *ba*, *dc* and *da* of the parallelogram, and draws segments connecting these intersections with the vertices of the parallelogram. Gerard combines these actions, which are recognized by the system as forming part of the line of the solution tree ''equivalent due to complementary dissection rules'', with generally false deductive actions on the comparison of segments. This combination of recognized actions means that Gerard's performance never has more than 3 actions together that are not recognized and the agent tutor therefore does not send any new message.

Gerard's graphic actions are always the same in nature—the identification of points and drawing of segments. In contrast, in his deductive actions we have observed a significant evolution, from a simple comparison of sides and segments in general (actions 47, 48, 55 and 56) to the comparison of multiples of segments (actions 57 and 58), and finally, comparison of areas as products of segments (actions 68 and those following) (Table 3).

With the validation of action 69, Gerard considers the argumentation to be sufficient and finishes the solving process. He saves the entire solving process in his case history and clicks on the solution validation button. The system concludes that Gerard's solution has not been fully argued. Despite having found the result of the problem—area *adm* = area *amb*—the number of recognised actions is 27%, not the 30% that the agent tutor requires.

To summarize this episode, we should mention that Gerard carries out actions related to the comparison of sides and segments only after observing the corresponding message from the previous episode; this means that there was a significant evolution in his

Table 3 Part of the episode of implementation in which the evolution of the deductive sentences entered by Gerard are shown

47	*Gerard*	*Deduction: line dm = line am*
48	*Agent mediator*	*line dm = line am: false*
...	...	... ...
55	*Gerard*	*Deduction: line ia = line in*
56	*Agent mediator*	*line ia = line in: false*
57	*Gerard*	*Deduction: line ia * 4 = line da [after much doubt)*
58	*Agent mediator*	*line ia * 4 = line da: false*
...	...	... ...
68	*Gerard*	*Deduction: line da * line za = line ba * line ia*
69	*Agent mediator*	*line da * line za = line ba * line ia: true*
70	*Gerard*	*Saves*
71	*Gerard*	*Definitive deduction: line da * line za = line ba * line ia*
72	*Agents mediator and tutor*	*line da * line za = line ba * line ia: true*
Message: This is not the correct solution		

comparisons. He went from making simple comparisons to comparisons of products of segments. Furthermore, we can see that although there is a combined use of the graphic and deductive areas, the former played a greater role, in that the graphic actions were subject to their use in the deductive area. An argumentative process in relation to the agent tutor was thus begun.

4 Theoretical underpinnings of the development of AgentGeom

In this section we analyse the main theoretical underpinnings of the development of AgentGeom. We begin by establishing the notion of appropriation. We shall continue by showing the discursive characteristics of the system; following this, we show the didactic landmarks with regards to the development of autonomy in a student and finish by locating the system in relation to other projects.

4.1 The notion of appropriation

The messages in AgentGeom follow the model of interactions from Cobo and Fortuny (2000), and Kieran (2001), as well as the idea of *appropriation* Moschovich (2004). The model of interaction allows the evaluation of the progression of cognitive and heuristic abilities in a collaborative solving process. These interactions are compatible with the dialectical model of Lakatos (1976), in the sense that they can be related to the formulation of the conjecture, the process of argumentation, the organization of knowledge or the mode of reasoning (e.g., treatment by congruence or equality of measurements). The interpretation of these interactions makes it possible to identify, in a solving process between two pairs, whether the cognitive and heuristic production are of the same order, i.e., if the interlocutors reason on the same objects, contribute towards the formulation or the proof of the same conjecture, and if their initiatives or their reactions are divided mutually in the argumentative process. Appropriation is a central neo-vygotskian concept that has been used to describe how learning is mediated by interaction with others and how children learn when adults guide or teach them (Moschkovich 2004). This sociocultural perspective implies, primarily, that learning mathematics is viewed as a discursive activity (Forman 1996).

4.2 About discourse

In order to simulate conversation and allow the input of solutions, AgentGeom uses a structural reasoning approach based on the notions of Duval's discursive expansion (1995) and Richard's graphic expansion (2004b). For Duval, the reasoning process is structured on reasoning steps—which we generically refer to as ''inferences''—in a logical process (e.g., deductions, Aristotelian syllogisms) or dialogue (e.g., semantic inferences, discursive inferences). According to Richard, these steps can mobilize several registers of semiotic representation within the same inference that, therefore, define the nature of the inference, but which above all make functional the joint use of the figural register and of discourse. This is why the student creates graphic propositions (elements of figures such as a point, a segment, a straight line, etc.) and discursive propositions (e.g., the writing of an equality). These *meaningful actions*—we could have said *operators* in the sense of Balacheff and

Margolinas (2005)—are turned towards the writing of propositions for the implementation of the reasoning of the student, that is the verbal expression of meaningful actions. In consequence, the student works on the basis of these propositions so that they can be used as premises to produce inferences, using a menu that contains the rules of deduction.

AgentGeom depends on a solution tree that is constructed a priori, (Fig. 6) and the algorithm that determines the messages to return (Fig. 4). The solution tree is constructed and modified by an expert (teacher, educational software technician), who must make choices about which hypotheses and rules of inference are acceptable in the context of a problem. For example, if the expert decides to permit both true and false hypotheses in a problem, where the accepted rules of inference are cognitively valid, this authorizes the introduction of contradictions. But even if there is a contradiction, the algorithm can be placed in comparison with the solution tree and generate an adapted message. In fact, if the structural approach allows for the system to be equipped with a complete set of messages by default (Aïmeur et al. 2005), it follows that the expert must be able to adapt the content of each message wherever needed. In this way, following the introduction of a contradiction—represented in the graph by an inference that does not move towards the conclusion—the messages can be reformulated and sent taking into account the contradiction in question. When a student writes or chooses a proposition and the agent tutor returns a message, we can say then that they collaborate in the solution process. In other words, both are dealing with the same content, even the same inference.

The construction of an element of a figure or a deduction determines the meaning of the *discursive-graphic unit* (see Richard 2004a, 2004b) on which the reasoning of the student is based. AgentGeom's messages, which stick to the internal logic of the problem and its resolution (see the following section), react to a configuration of the preceding meaning units, which reveal the strategy of the student in the solving process (search for a conjecture and proof).

4.3 Development of autonomy

In AgentGeom, the question of the autonomy of the student carries a double logic: the logic of the solution tree, which is based on the strategy of the pupil in a given situation, and the logic of the didactic contract. From the theory of didactic situations, we know that: ''*the modern conception of teaching asks the teacher to provoke the adaptations expected in the student through a wise choice of problems that he poses. These problems are chosen in such a way that the student can accept them, act on them, discuss them, and reflect on them, allowing him to evolve in his own way*. (...) *The student is fully aware of the fact that the problem has been chosen for him so that he may acquire new knowledge, but he must also realize that this knowledge is entirely justified through the internal logic of the situation and that he can attain it without relying on didactic reasoning* (...). *Not only can he, but he must, as this knowledge will only have truly been acquired if he is able to put it into action himself* (...)'' (Brousseau 1997).

It ensues from this that the student can only accede to autonomy under the condition that he first respects the internal logic of the situation. If this notion of ''internal logic'' seems vague, it is due to the fact that mathematical models transposed onto learning situations must also obey the distinctive logic of the didactic contract in which the situation arises. This is the case when the geometric knowledge conveyed in a particular class is never formalized, but its meaning renders it operational in the situation. Therefore, the autonomy of the student is not only a quest for satisfiability in a completely mathematical internal logic, which presumes, perhaps, an interpretation in an absolute mathematical model, but also figures within the constraints of a didactic contract that must be the point of reference for both teacher and student alike.

Within a didactic contract, the help that an agent tutor can offer through the messages must occur on several levels. Firstly, the level that affects the constituents of the didactic relationship (milieu, contract, role of the agent) and that is generically bound to the devolution of the problem (in the Brousseau sense, 1997). However, it is possible that the student refuses to, avoids, or doesn't solve the problem, despite having read the first message. When the student loses motivation in the resolution process, the teacher has the social obligation to help the student and even sometimes to justify having posed a question that is too difficult. It must be remembered that our multi-agent system is a complementary approach to regular teaching; it isn't meant to replace it. On the other hand, when the student experiences difficulties in the solving process itself, it is then necessary to change levels. In this way, the agent tutor can take over, firstly through mathematical-type suggestions (e.g., ''you deduce a term according to a property, but it based on which assumption?''), then through suggestions that directly affect the knowledge engaged in the solving process. A description of the chosen levels can be found in Sect. 3 on an analysis of the student's cognitive progress by means of his/her or her interactions.

4.4 AgentGeom and related systems

AgentGeom has a predecessor in Cabri-Euclide, a environment developed by Luengo (1999) and on which the Baghera project was partly based. The Baghera project was developed in the Leibniz laboratory in Grenoble (Laboratoire Leibniz 2003). The Baghera project, like our tutorial system, incorporates three principles, which we regard as being fundamental in the design of educational environments based on collaboration between human and artificial agents:

- The computer was considered as an autonomous device that conceived teaching only in its instructional function (Balacheff 2000).
- We acknowledge that students must possess some knowledge of the mathematical entities and procedures that are being referred to when they are learning. However, in any given learning situation, the students knowledge will only be partially coherent with the that of other agents looking at the same situation. That leads to the fact that we regard education as the result of a complex process that emerges from interactions between agents having different and complementary abilities, and not as the result of the action of one isolated strategy or the accomplished goal of one isolated agent (Webber et al. 2002).
- Finally, just like Baghera, AgentGeom is conceived as a multi-agent, diagnostic tutorial system that can identify students' implemented knowledge starting from their interactions with the system. This allows the system to adapt to the cognitive characteristics of the students and to the evolution of their mathematical knowledge.

However, there are fundamental differences between AgentGeom and Baghera. In AgentGeom, the checking of the accomplishment of the students' activities is carried out by the system every time he or she performs any action. We subscribe to the idea that a learning environment should provide, in real time, support for exploration processes, adapted to the individual needs of each student (Bunt and Conati 2002). We conceived our tutorial system so that it considers and controls instantaneously all the significant actions of the student in the solving process. On the one hand, it requires student to make their own discursive and cognitive contributions explicit. It also shows them, at their request, messages that direct their initiatives while still maintaining the collaborate character of the interaction. This approach comes with differentiated support for each student according to the evolution of the solving process, and it is adapted to their ability. This gives an emergent character, in the sense that an interaction between a student and AgentGeom cannot be predicted by describing the components of the system with precision.

Predecessors of AgentGeom can also be found in the literature on Intelligent Tutor Systems for the learning of geometry. These systems considerer only aspects of formal geometry (Koedinger and Anderson 1993); that is, they are only concerned with the algorithmic manipulations whereby given assumptions and axioms can be manipulated to arrive at a given conclusion. However, this model doesn't capture the full nature of geometry as it is faced by students in the classroom. In fact: ''the machine-generated construction is often different from the one asked of the student (who is usually working in a cognitive geometry; i.e., within a certain context of definitions, theorems, and exercises which do not include all of geometry) and is useful as a metaphor relating Euclidean geometry to analytic geometry and linear algebra'' (Allen and Trilling 1997). In comparison, cognitive tutors like AgentGeom organize problem situations bearing in mind only the references of the curricular objectives within an educational system.

5 Discussion and Conclusions

When a student collaborates with AgentGeom to jointly ''solve'' a problem, changes are generated in the way students learn. In our research, these changes are expressed in terms of new appropriations, in the way they produce deductive sentences, and in how they understand the meaning and the need for mathematical proofs. Moreover, these changes can be seen in transformations in the students' performance as they evolve towards the use of an argumentative process based on the shared use of graphic and deductive areas of the artificial tutoring system. We have identified three types of interactions with AgentGeom in Gerard's problem solving process for the parallelogram. Their characteristics are described below:

- The student's taking of the initiative and use of proactive actions that revolve around the use of AgentGeom's graphic area is characteristic of what we called exploration/analysis episodes. In this kind of episode, it could be said that Gerard only uses the graphic area as a simple drawing tool, in that he detaches the graphic constructions from those that are deductive (he does not use them). We interpret Gerard's ''graphic'' initiative as stemming from, firstly, the implicit need to apply the formula of the area of the triangle as the only resource for reaching a solution for the problem, and, on the other hand, from the reproduction of the manner of solving problems using a paper and pencil, in which he never uses argumentative processes such as those used in the deductive area. Throughout this episode, Gerard misses the opportunity to go more deeply into the strategy because he does not know how to interpret the non-verbal forms of communication (in this case the silence of the agent tutor) and because he does not request any messages that could orient him towards a solution. This lack of understanding is the starting point in the evolution of the solving process, in which the mathematical contents used by Gerard always are related to drawing of perpendicular lines and segments related to the height of the triangle.

- The second type of interaction is the episode that we have called *analysis*. This episode is essentially reactive, because AgentGeom causes Gerard's actions. This can happen indirectly, such as when Gerard observes the word ''perpendicular bisector'' in the procedures window, or when he requests a message. Or it can happen directly, such as when Gerard reacts to the message that the agent tutor sends to him about the comparison of sides. The agent tutor's performance, as regards the sending of messages, should be highlighted. To help Gerard get beyond the impasse in which he finds himself after receiving the message about comparing the two sides, a new message is selected, at Gerard's request, for a change of strategy and a higher level of information, which helps him get beyond the impasse In this episode, Gerard has started to benefit from the agent tutor's potential, namely, the use of the graphic and deductive areas in an argumentative process, which for the time being, has led him to establish a conjecture as to the result of the problem that he is trying to solve. Gerard therefore finds himself in an ideal situation to solve the problem. He knows what he wants to do—to test the conjecture—and the way in which he wants to do it, i.e., by stringing together a series of sentences, even though he still does not know their content.
- This learning about the idea of mathematical proof becomes evident in Gerard's third type of interaction with the agent tutor, which takes place in the *implementation* episode. We classify the implementation episode as reactive, because Gerard's actions are based on observation of the message about comparing sides, and Gerard's actions with mathematical content revolve around this observation. The agent tutor and Gerard, each carrying out their own role, continue to contribute to the development of the solving process. Gerard continues to generate recognized graphic actions following the strategy of ''equivalent due to complementary dissection rules'', at the same time as alternating deductive sentences, which are often false. Meanwhile, the agent tutor remains silent, only reacting to the grammatical errors in Gerard's sentences. Gerard now begins to understand this behaviour and does not get stuck. He continues to develop his strategy, with more and more enthusiasm, which is shown by his need to produce new deductive sentences about objects that he creates, even before creating them.

We interpret Gerard's conduct as confirming our expectations at the conclusion of the prior episode, with regard to a sense of even greater integration of graphic-deductive reasoning in benefit of the argumentative process. In the same way, we understand the evolution of Gerard's conduct towards the appropriation of the idea of a mathematical proof as being the result of his interaction with the agent tutor. We can say that both of them have contributed to the construction of the meaning of this concept.

We can highlight, on the one hand, the aspects related to the technological characteristics of AgentGeom, and on the other, its pedagogical application when helping students in the problem solving process:

- As regards its technological characteristics, AgentGeom is a multi-agent tutorial system that combines two basic functions in any educational system—it is open and allows attention to diversity, as it provides the necessary mechanisms so that the teacher can: a) broaden the series of problems; b) manage the system; c) create new problems; d) assign them to his/her pupils according to their cognitive characteristics; e) examine the effectivity of their processes of resolution; and finally f) modify the system of messages that she can send to each pupil in each problem according to the strategies that have been chosen. Moreover, the AgenGeom can check and verify all the actions carried out by the pupils instantaneously.

- As regards its pedagogic collaboration with a pupil in the solving of problems, we can say that AgentGeom has generated changes in the pupil's learning. In our research, these changes have been related to appropriations, of the form of producing deductive sentences and understanding both the meaning and the need of the mathematical proofs in the solving of geometry problems. Furthermore, these changes were evident in transformations in the performances of the pupil who have evolved from the graphical and deductive areas of the AgentGeom towards a use of the argumentative process based on a shared use.

The AgentGeom tutorial system has novel features that distinguish it from similar systems, as we have mentioned above. Despite this, there are two aspects of AgentGeom that while advantageous in principle, might in practice might be disadvantageous to student learning. These problems have to do with the lack of autonomy that the use of AgentGeom may cause. More specifically,

- A continuous use of AgentGeom may cause students to have a lack of initiative when they face solutions without the help of AgentGeom. This disadvantage could be alleviated by making the system give increasingly staggered help messages.
- Continued use of AgentGeom may make students dependent on the system's validations. In fact, when they are not using AgentGeom, we have noted that students miss the speed and security given to them by the system when validating the sentences they input. In the classroom, they should normally carry out these validations. Indeed, teachers, if they want to develop their students' autonomy, must encourage them to establish the veracity of the statements input. We have tried to be very sensitive to the help given by the system to students. The system only shows messages with mathematical information when this is absolutely necessary. Despite this, it is possible that this dependence occurs. To reduce this likelihood, we can make the following recommendations: i) alternate use of AgentGeom with problems that students have to solve in class, individually or in groups; ii) use the debating forum provided by the system, during or after problem solving with AgentGeom; iii) modify of the AgentGeom validation system, so that sentences input by the student are not validated until the student writes the property, the definition and the axiom on which they are based and the system is able to recognize them.

In conclusion, we can say our work seeks to discover which geometrical proof abilities students can develop and how they can be actively transformed after the suitable adjustment of the messages that we have mentioned above.

Acknowledgements The authors are indebted to the editors and the anonymous reviewers for their helpful comments. Dr. Pedro Cobo has contributed to the development of this research thanks to a study license granted by Departament d'Ensenyament de la Generalitat de Catalunya (DOGC núm. 3926 of the 2003-07-16).

References

Aïmeur, E., El-Khoury, S., Fortuny, J. M., & Richard, P. R. (2005). An open architecture to improve the mathematical competences in high school. *Actes de la ED-media world conference of association for the advancement of computing in education* (pp. 3395–3402). Montréal.

Allen, R. J., & Trilling, L. (1997). Dynamic geometry and declarative geometric programming. In J. R. King, & D. Schattschneider (Eds.), *Geometry turned on. Dynamic software in learning, teaching, and research*. (pp. 193–197) Washington: MAA Notes Series.

Balacheff, N. (2000). Teaching, an emergent property of eLearning environments. In *Proceedings of the international conference on intelligent tutoring systems 2000*. Nice. Retrieved at http://www-didactique.imag.fr/Balacheff/TextesDivers/IST2000.html.

Balacheff, N., & Margolinas, C. (2005). Modèle de connaissances pour le calcul de situations didactiques. In A. Mercier, & C. Margolinas (Eds.), *Balises pour la didactique des mathématiques* (pp. 75–106). Grenoble: La Pensée sauvage.

Brousseau, G. (1997). *Theory of didactic situations*. Dordrecht: Kluwer Academic Publishers.

Bunt, A., & Conati, C. (2002). Assessing effective exploration in open learning environments using bayesian networks. In *Proceedings of the international conference on intelligent tutoring systems 2002*. Biarritz. Retreived at http://www.cs.ubc.ca/%7Econati/my-papers/its2002BuntConati.pdf.

Cobo, P., & Fortuny, J. M. (2000). Social interactions and cognitive effects in contexts of area-comparison problem solving. *Educational Studies in Mathematics, 42*, 115–140.

Cobo, P. (1998). Análisis de los procesos cognitivos y de las interacciones sociales entre alumnos (16–17) en la resolución de problemas que comparan áreas de superficies planas. Un estudio de casos. *Doctoral dissertation*. Universitat Autònoma de Barcelona.

Cobo, P. (2004). Disseny d'agents pedagògics intel·ligents per millorar les competències estratègiques de l'alumnat en la resolució de problemes de matemàtiques. *Memoria inédita de la llicencia de estudios concedida por el Departament d'Ensenyament de la Generalitat de Catalunya* (DOGC, núm. 3926 de 16-7-2003).

Duval, R. (1995). *Sémiosis et pensée humaine: registre sémiotique et apprentissages intellectuels*. Berne: Peter Lang.

Forman, E. (1996). Learning mathematics as participation in classroom practice: implications of sociocultural theory for educational reform. In L. Steffe, P. Nesher, P. Cobb, G. Goldin, & B. Greer (Eds.), *Theories of mathematical learning* (pp. 115–130). Mahwah, NJ: Lawrence Erlbaum Associates.

Kieran, C. (2001). The mathematical discourse of 13-year-old partnered problem solving and its relation to the mathematics that emerges. *Educational Studies in Mathematics, 42*, 115–140.

Koedinger, K., & Anderson, J. R. (1993). Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design. In S. P. Lajoie, & S. J. Derry (Eds.), *Computers as cognitive tools* (pp. 15–46). Hillsdale, NJ: Erlbaum.

Laboratoire Leibniz (2003). Baghera assessment project: Designing an hybrid and emergent educational society. In S. Soury-Lavergne (Ed.), *Rapport pour la commission européenne, Programme IST*. Grenoble: Les Cahiers du Laboratoire Leibniz n° 81.

Lakatos, I. (1976). *Proof and refutations*. Cambridge: Cambridge University Press.

Luengo, V. (1999). Cooperative agents to learn mathematical proof. In: P. Kommers, & G. Richards (Eds.), *Proceedings of world conference on educational multimedia, hypermedia and telecommunications* (p 1632). Chesapeake, VA: AACE.

Matsuda, N., & Okamoto, T. (1998). Diagramatic reasoning for geometry ITS to teach auxiliary line constuction problems. In *Proc. of the international conference on intelligent tutoring systems* (pp. 244–253). Berlin: Springer.

Moschkovich, J. (2004). Appropriating mathematical practices: a case study of learning to use and explore functions through interaction with a tutor. *Educational Studies in Mathematics, 55*, 49–80.

Rabardel, P. (2001). Instrument mediated activity in situations. In: Blandford A., Vanderdonckt J., Gray P. (eds.), *People and computers XV—interactions without frontiers* (pp. 17–30). Berlin: Springer-Verlag.

Richard, P. R. (2004a). *Modélisation du comportement en situation de validation*. Berne: Peter Lang.

Richard, P. R. (2004b). L'inférence figurale: un pas de raisonnement discursivo-graphique. *Educational Studies in Mathematics, 57*(2), 229–263.

Schoenfeld, A. H. (1985). *Mathematical problem solving*. Orlando: Academic Press.

ScreenFlash (2002). Retrieved at http:// www.unflash.com.

Webber, C., Bergia, L., Pesty, S., & Balacheff, N. (2002). The Baghera project: a multi-agent architecture for human learning. In *Proceedings of the workshop multi-agent architectures for distributed learning environments (AIED2001)* (pp. 12–17). San Antonio.

Winston, P. H. (1992). *Artifical intelligence*. New York: Addison Wesley.