S. L. Wong et al. (Eds.) (2010). Proceedings of the 18th International Conference on Computers in Education. Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.

Drawing Dynamic Geometry Figures with Natural Language

Wing-Kwong WONG[a], Sheng-Kai YIN[b], Chang-Zhe YANG[c]
[a]*Department of Electronic Engineering*
[b]*Graduate School of Engineering Science & Technology*
[c]*Graduate School of Computer Science and Information Engineering*
National Yunlin University of Science & Technology
Douliou, Yunlin, Taiwan
{wongwk, g9310811, g9617727}@yuntech.edu.tw

Abstract: This paper presents a tool for drawing dynamic geometry figures by understanding the texts of geometry problems from common textbooks. With the tool, teachers and students can construct dynamic geometry figures on a webpage by inputting a geometry problem in natural language. First a knowledge base for understanding geometry problems is built. With the help of a knowledge engineering tool InfoMap, geometric concepts are extracted from an input text. The concepts are then used to output a multi-step JavaSketchpad script that constructs a dynamic geometry figure step by step. Finally, the system outputs the script as a HTML document that can be read and visualized with an internet browser.

Keywords: natural language understanding, dynamic geometry, JavaSketchpad, geometry education, e-learning

1. Introduction

The National Council of Teachers of Mathematic has published two important documents on K-12 mathematics curriculum: Curriculum and Evaluation Standards for School Mathematics [13], and Principles and Standards for School Mathematics [14]. The latter focused more on the skills of writing formal proofs of geometry [6]. Furthermore, mathematicians and educators agree that the ability in writing geometry proofs involve important skills that are difficult to learn [7] [16].

Some computer programs are used in teaching geometry in school. In fact, using geometry software also increases the motivation of students in learning geometry. Some popular programs include Geometer's Sketchpad (GSP), Cabri Geometry II, Geometry Expert, and Cinderella’s Café. These programs share one common focus on dynamic geometry (DG) figures. In a DG figure, students can drag a geometry object such as a vertex of a triangle and change the figure dynamically while preserving the given conditions of the figure and all geometric invariants, which are the consequences of the given conditions. Thus these programs are commonly used to demonstrate geometry theorems.

Dynamic geometry can also be used by students to discover conjectures about a figure in some given conditions. Students can explore various configurations of a DG figure and try to discover conjectures about the figure on their own. For some learning activities in classroom, students are given work sheets to fill in measurements of some properties of geometric objects and write down conjectures they discover in a DG figure. There are two common problems in such activities. First, teachers and students have to learn to use the DG software and the learning process can be difficult and time consuming. Second, there is little record on how students manipulate the dynamic figures in order to conclude their findings. Both problems make it difficult to evaluate the effectiveness of learning from DG systems.

To address the above two problems, we build an online dynamic geometry system that can “understand” common, basic geometry problems. With this system, the teachers and students can construct DG figures on a webpage for their learning activities, such as making geometry conjectures and proving a theorem. Furthermore, this tool can help researchers design user interface that collects data of students' interactions with DG figures. Based on the data, researchers can get more insights on how students make inferences from their interactions with the dynamic figures.

2. Background

Studies indicated that when students explored conjectures in a dynamic geometry environment (DGE), they could explain the formal proof they wrote based on their experiences in the exploration [4]. Furthermore, students would strengthen their beliefs in the geometry conjectures they made from their observation of the changes of dynamic figures in DGE [1] [2]. In DGE, students can notice the variation and invariance of conditions in a dynamic figure, deepening their understanding of geometry theorems [8]. Some studies also found that object dragging in DGE could reduce the gap between DG experimentation and the generation of theorem proving ideas by learners [10] [4] [3]. When DGE is used in some learning activities, students need both basic geometry knowledge and skill of working with DG tools such as GSP. Sometimes, they need to add geometric components to a dynamic figure. This can be an obstacle to students who are not expert users of DG tools, reducing the effectiveness of learning in DGE [15].

Lees and Cowie [9] proposed an enquiry system for training students to learn UNIX commands. The system provides a natural language interface for learners so that they can learn UNIX commands by themselves. Li and Chen [11] proposed a Chinese enquiry system about fundamental knowledge of computer. The system uses a Linguist String Parser to understand the question inputted by a learner, and then outputs an appropriate answer. A model is proposed to simulate procedural knowledge of basic arithmetic operations [12]. The model helps teachers design appropriate curriculum and teaching strategy from the records of students solving arithmetic problems. The model is used in an intelligent tutoring system that can accumulate and reproduce the knowledge from teachers and students and help teachers build a good learning map for students. The system LIM-G [17] is used to understand geometry word problems and help elementary school student comprehend geometry word problems, which are about the area or circumference of a geometric object. After a student inputs a geometry problem to LIM-G, the system understands the text using a pre-built geometry knowledge base and constructs the figure of the problem.

In this study, we help to make DGE more accessible to teachers and students. We propose to build a system that can draw dynamic geometry figures by understanding texts of geometry problems. In this way, teachers and students are not required to spend so much time in constructing figures in DGE. Moreover, if the system is available on a website, then there is no need to install any expensive commercial software in school, making DGE more accessible to schools in poor school districts.

3. Core technologies of the system

3.1 System architecture

There are three main components in this system. The first is the knowledge inference engine InfoMap. When a user enters a geometry problem in natural language, InfoMap analyzes the problem and extracts the attributes of the geometric concepts in the problem. This information is sent to the second component of the system. This component generates a JSP script that draws a DG figure of the problem, which can then be manipulated by the user to explore various conjectures. The third component outputs the drawing script of JavaSketchpad as a HTML document that can be loaded by any web browser to display the DG figure. Figure 1 shows the system architecture with the three components.

Figure 2 shows a snapshot of the user-interface in a web browser. A user can input the geometry problem in Chinese and simple mathematical symbols in the text area at the bottom. The canvas at the top displays the dynamic figure according to the input problem. The figure is drawn with a JavaSketchpad script embedded in a HTML document.

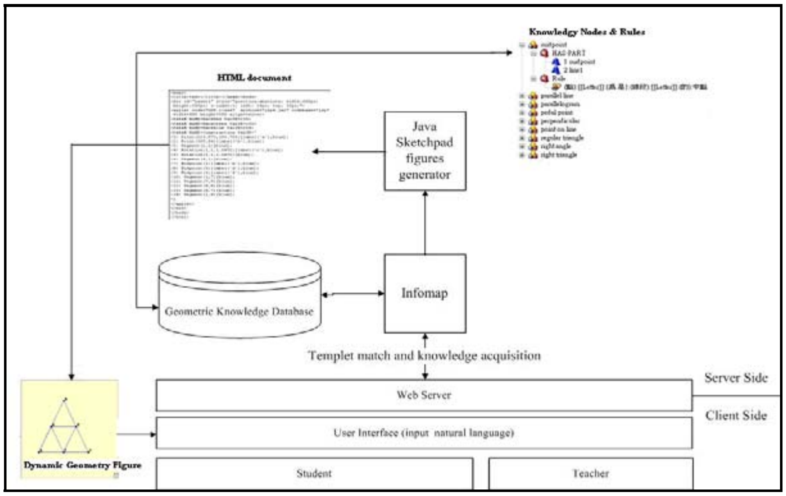

Figure 1. The system architecture

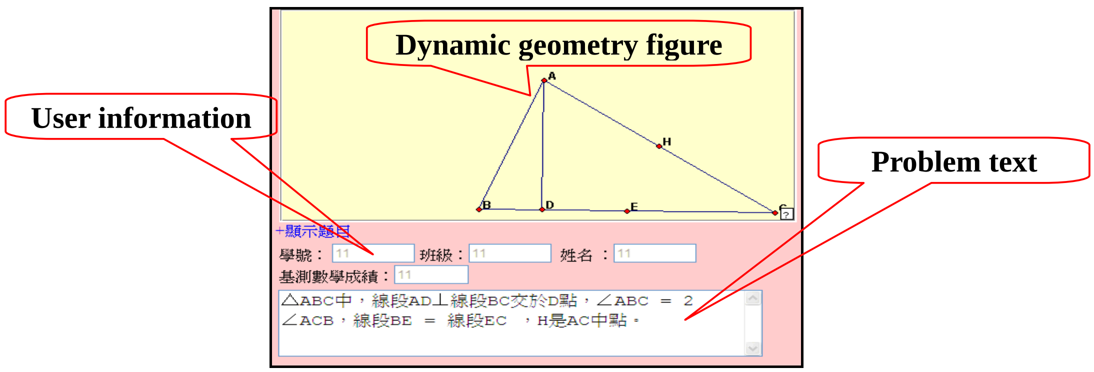

Figure 2. A snapshot for the user-interface

3.2 Infomap and knowledge base of geometry concepts

InfoMap is a knowledge engineering tool provided by the Intelligent Agent System Lab, Institute of Information Science, Academia Sinica. InfoMap is an ontology-based system for knowledge representation and template matching [5]. InfoMap works as an agent by understanding texts in any domain and can answer questions about them, if the needed domain knowledge is provided. In a knowledge base in InfoMap, nodes represent geometry concepts and each template of a node specifies the syntax of sentences that use the concept of the node. Templates are matched to input sentences and concepts can be extracted from the sentences.

Figure 3 shows part of the knowledge base, which includes many concepts, e.g., midpoint, pedal point, intersection, triangle, isosceles triangle, parallelogram, parallel line, point on line. In this study, we have built more than 50 concept nodes of geometry. A concept node is also a knowledge frame, which includes a rule node and two attribute nodes. The rule node generally includes multiple templates, which describe the syntax of sentences about the concept. The HAS-PART node specifies the component nodes that make up the concept. The component nodes can store the concepts and their names that are extracted from an input sentence.

Suppose the user inputs a Chinese sentence meaning “Point A is the midpoint of segment BC” or “A is the midpoint of BC”. The sentence is matched against the InfoMap template of midpoint, and the node “midpoint” is triggered. InfoMap will extract the component concepts of midpoint point A and segment BC, and then label the component “midpoint” as “A” and component “line1” as “BC”. The template for an equivalent English sentence is “(Point) [[Letter]] is (the) middle point (of) (segment) [[Letter]]”. So the sentences “Point A is the midpoint of segment BC” and “A is the midpoint of BC” both match the template of the "midpoint" concept. One matched result is shown in Table 1.

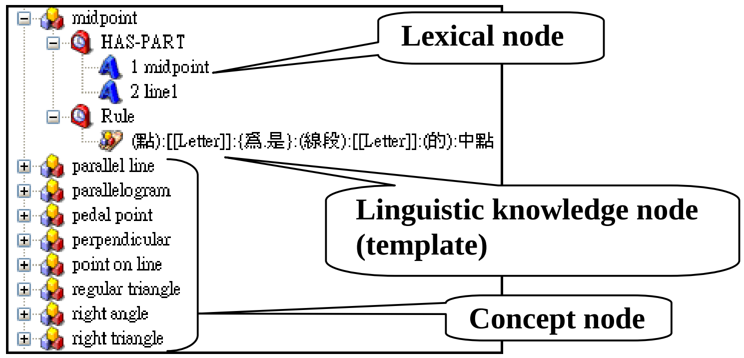

Figure 3. Knowledge base of geometry concepts

Table 1. Result of template matching for one sentence

Geometry proof description	Point	A	is	the	middle point	of	segment	BC
Template	(Point)	[Letter]	is	(the)	middle point	of	segment	[[Letter]]

3.3 JavaSketchpad and text understanding

JavaSketchpad (JSP) is a computer program with which authors publish DG figures of Geometer's Sketchpad as a Java applet embedded in a HTML file so that users can interact with the figures with a web browser on the internet [3]. Geometer's Sketchpad from Key Curriculum Press is a DGE that runs on a personal computer. Moreover, instructors can publish interactive, DG content in learning activities so that students can participate in the activities over the internet. GSP supports a web solution by publishing a HTML document embedding a Java applet containing the JavaSketchpad script into the document (Figure 4).

Figure 4. JavaSketchpad Script and HTML document

There are two different methods for parsing an input geometry problem text. The first method parses all sentences of a text at a time [17]. This method is not flexible as it requires many templates of possible combinations of the sentences in the text. A better method is to parse one sentence at a time and then integrate the results for all sentences. Consider the following problem: *"Consider parallelogram ABCD. The point E is the midpoint of segment AB. F is the midpoint of segment CD. Prove the length of segment DE is equal to the length of segment FB."* This text is segmented into four sentences. Each sentence is matched in InfoMap and the concepts of the sentences are extracted and mapped to JavaSketchpad command, which is then generated. The system needs to map the concepts of a sentence into one or more JavaSketchpad (JSP) commands in order to draw the concepts. The example of *"parallelogram ABCD"* is mapped to the JSP commands in Figure 4. Since there is no parallelogram command in JSP, we have to design a specific multi-step script for drawing a parallelogram.

4. Conclusion

Dynamic geometry environment such as Geometer's Sketchpad is recognized as a tool with great potential educational value. In a DGE, student can see some invariant results, among other varying conditions, under given geometric premises. Unfortunately, it can be difficult for instructors and students to use tools in a DGE to construct dynamic figures. We propose to address this problem by automatically drawing dynamic figures from input problem texts. A system is built for this purpose, using a knowledge base of basic geometry concepts and an inference engine InfoMap to map problem texts into JavaSketchpad scripts. A JavaSketchpad script embedded in a HTML document can be viewed by a browser on the internet.

Acknowledgement

This study is supported by the National Science Council, Taiwan (NSC 98-2511-S-224-005-MY3, NSC 98-2511-S-224-004-MY2).

References

[1] De Villiers, M., 1996, “Some Adventures in Euclidean Geometry”, University of Durban-Westville, South Africa.
[2] De Villiers, M., 2003, “Rethinking Proof with Geometer's Sketchpad 4”, Emeryville: Key Curriculum Press, USA.
[3] Furinghetti, F. & Paola, D., 2003, “To produce conjectures and to prove them within a dynamic geometry environment: a case study”, In N. A. Pateman, B. J. Dougherty, J. T. Zilliox (Eds.), Proc. of the joint meeting PME 27 and PMENA, Vol. 2, pp. 397-404.
[4] Hoyles, C. & Healy, L., 1999, “Linking Information Argumentation with Formal Proof Through Computer-Integrated Teaching Experiences”, In Zaslavsky (Ed.), Proceedings of the 23nd conference of the International Group for the Psychology of Mathematics Education, Haifa, Israel.
[5] Hsu, W. L., Wu, S. H. & Chen, Y. S., 2001, “Event identification based on the information map – INFOMAP”, Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference, Tucson, Arizona, USA, pp. 1661-1672.
[6] Knuth, Eric J., 2002, “Teachers’conception of proof in the context of secondary school mathematics”, Journal of Mathematics Teacher Education, 5(1), pp. 61-88.
[7] Koedinger, K., 1998, “Conjecturing and argumentation in high-school geometry students,” In R. Lehrer & D. Chazan (Eds.), Designing Learning environments for developing understanding of geometry and space, Mahwah, NJ： L. Erlbaum, pp. 319-347.
[8] Laborde, C., Kynigos, C., Hollebrands, K., & Strasser, R., 2006, “Teaching and learning geometry with technology”, In A. Gutierrez & P. Boero (Eds.), Handbook of research on the psychology of mathematics education: Past, present and future, Rotterdam: Sense publishers, pp. 275-304.
[9] Lees, B., & Cowie, J., 1996, “Applying natural language technology to the learning of operating systems functions”, Proceedings of the 1st conference on integrating technology into computer science education, pp. 11-13.
[10] Leung, A. & Lopez-Real, F., 2003, “Properties of tangential and cyclic polygons: an application of circulant matrices”, International Journal of Mathematical Education in Science and Technology. 34(6), pp. 859-870.
[11] Li, P. Y., & Chen, J. D., 1988 “A Computer Training Tool Using Chinese Natural Language”, IEA/AIE, Vol. 2, pp. 899-904.
[12] Lu, C. H., Wu, S. H., Tu, L. Y., & Hsu, W. L., 2005, “Ontological Support in Modeling Learners' Problem Solving Process,” Journal of Educational Technology & Society, 8(4), pp. 64-74.
[13] National Council of Teachers of Mathematics, 1989, “Curriculum and Evaluation Standards for School Mathematic”, Reston, VA: NCTM.
[14] National Council of Teachers of Mathematics, 2000, “Principals and standards for school mathematics”, Reston, VA: NCTM.
[15] Talmon, V, & Yerushalmy, M., 2004, “Understanding Dynamic Behavior: Parent-Child Relations in Dynamic Geometry Environments”, Educational Studies in Mathematics, Vol. 57, pp. 91-119.
[16] Whiteley, W., 1999, “The decline and rise of geometry in 20th century North America,” In John Grant McLoughlin (Ed.), Canadian mathematics education study group, St. John’s, NF： Memorial University of Newfoundland, pp. 7-30.
[17] Wong, W. K., Hsu, S. C., Wu, S. H., & Hsu, W. L., 2007, “LIM-G： Learner-initiating instruction model based on cognitive knowledge for geometry word problem comprehension”, Computers and Education Journal, 48(4), pp. 582-601.