DYNAMIC BEHAVIOR IN DYNAMIC GEOMETRY ENVIRONMENTS: SOME QUESTIONS OF ORDER

Varda Talmon and Michal Yerushalmy

University of Haifa, Israel

Dragging within Dynamic Geometry Environments is a tool for investigating relationships in geometric figures at both the perceptual and formal levels. It produces a Dynamic Behavior for each element in the construction. This paper reports on part of a larger experiment conducted to analyze the complexity involved in the understanding of Dynamic Behavior and to characterize Dynamic Behavior as part of the instrumental genesis of DGE. Among the many components of dragging, the article chooses the order or hierarchy of construction for its focus. We demonstrate the complexity involved by examining the design of three DG environments and analyzing a student's interaction with a DGE.

HIERARCHY OF DEPENDENCIES: BACKGROUND AND RATIONAL

Dynamic Geometry Environments (DGEs) offer two main tools: primitives, commands and macros used to create geometric constructions, and drag mode for manipulating them easily and continuously. By allowing users to move certain elements of the construction and to observe how others respond to the altered conditions, dragging produces a Dynamic Behavior (DB) for each element in the construction. DB refers to the degree of freedom of the dragged element and the response of the other elements, that is, changes and invariance.

The sequential organization of actions necessary to produce a figure in any Dynamic Geometry software introduces an explicit order of construction. In a complex figure this sequential organization produces what is, in effect, a hierarchy of dependencies as each part of the construction depends on something created earlier (Jones 2000). This hierarchy of dependencies is one of the main factors that determine DB within DGE. This order is often explicitly organized by terms such as parent and child within the primitives of the software, for example, The Geometer's Sketchpad or The Geometric Supposer.

The longer these tools are in use and under study the more we learn about their contribution to the learning of geometry but also about the obstacles they are liable to pose to such learning. (Chazan, & Yerushalmy, 1998; Goldenberg, & Cuoco, 1998; Jones, 2000; Healy, & Hoyles, 2001; Holzl, 2001; Straesser, 2001; Marioti, 2002). This study suggests that hierarchy of dependencies is one of the sources for obstacles. As Jones (2000) states: "For most users, order is not normally expected or does not even matter." We are interested in different facets of the instrumentalization process that are related to the use of DGE.

As a part of a larger study on the complexities involved in understanding DB, we are examining the connection between the order of construction and the DB of its elements. Here, to study these connections, we start by analytic investigation of the DB of two similar parallelograms. We proceed by analysis of interpretations of order of construction and DB made during an interview with a 14-year-old girl and by a brief comparative analysis of designs of DB in 3 different DGEs.

EUCLIDEAN DEFINITION AND ORDER OF CONSTRUCTION: AN EXAMPLE OF DYNAMIC BEHAVIOR

Behind every procedure of a geometric construction lies a geometric definition of the figure. The DB of the figure is derived from this definition. Based on the distinction between 'figure' and 'drawing' denoted by Parzysz (1988), Laborde (1993) argues that the DB can take the role of the written procedure. "The movement produced by the drag mode is the way of externalizing the set of relations defining a figure." (Laborde, 1993). But, although the Euclidean definition from which the construction is derived constitutes the central knowledge that needs to be taught, there are other relations that would undermine the one-to-one relationship between the definition and the DB.

For example, Figure 1 shows two different parallelogram constructions with the Geometric Supposer based on the same definition. Both are based on bisections of the diagonals. The difference between the two is the sequential organization of actions. The two parallelograms have different DBs, which produces different families of parallelograms.

Procedure 1 – parallelogram 1 ***Procedure 2 – parallelogram 2***

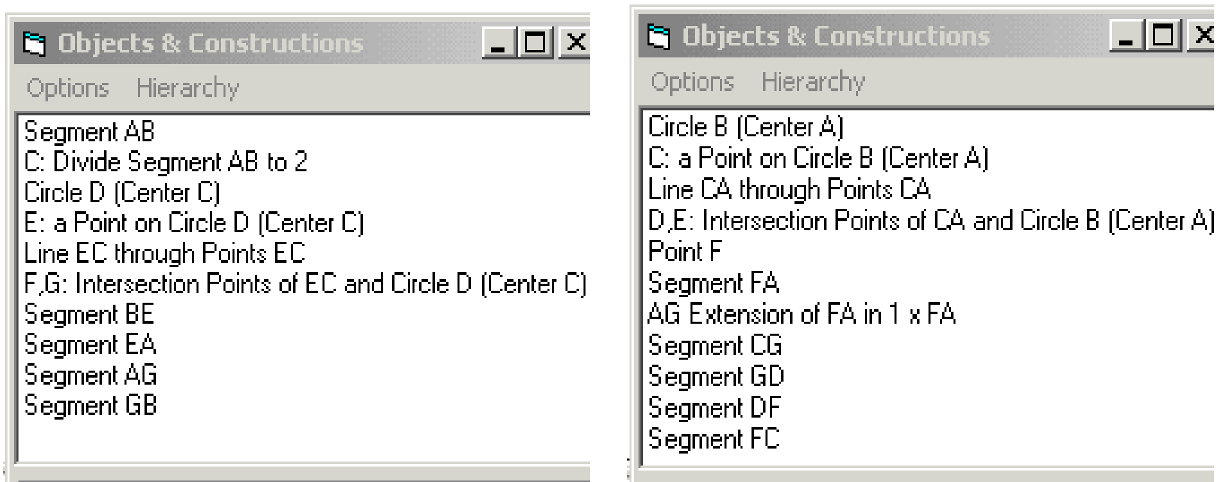


Figure 1: Two procedures based on the same definition

Comparing correspondently the DB of the four vertices of each parallelogram is the best way to demonstrate the differences. For example, we chose to follow the free vertex B of parallelogram 1 (A and B are the only free vertices and have the same DB) and the only free vertex F of parallelogram 2. The DBs are described in Figure 2. B and F behave differently and produce two different geometric families: one is parallelograms with diagonals of the same length, the other is parallelograms that share a common diagonal. Both constructions were built within the Geometric Supposer[1].

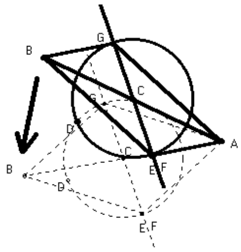
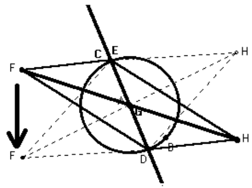
	Dynamic Behavior 1 parallelogram 1	**Dynamic Behavior 2 parallelogram 2**
Vertex 1	**Vertex B** 	**Vertex F** 
Degrees of freedom	Free	Free
Fixed/relocated elements	• Vertex A is fixed. • E and G move. • The circle moves as the center C moves.	• D and E are fixed. • The circle remains invariant (fixed). • Vertex H moves in a circular path of a changing radius FA.
Invariance under dragging	The diagonal; the radius of the circle is invariant.	One diagonal is fixed
Geometric family	Parallelograms having a diagonal of same length.	Parallelograms with a common diagonal.

Figure 2: DB of one vertex of two parallelograms based on the same definition

Although differences in DB that resulting from the different order or hierarchy may not be considered geometrically substantial, there are substantial visual and user-control differences. Our assumption is that these differences have an impact on the instrumental genesis (Marioti 2002) of DGE: What do users see, if anything at all, as the natural DB? Are they able to extract the important geometric commonalities from different DBs? Does visual control that even novice users are expected to benefit from suffer from a 'noisy' DB? Finally, are users aware of the effect of sequential organization on DB? We are exploring these questions with users. Below is an example.

---

1 Comparing the two constructions within other environments demonstrate the same idea

USER INTERPRETATIONS OF ORDER OF CONSTRUCTION

Inbar has been studying geometry with the Geometric Supposer for over a year,

including parallelograms. In the first part of the interview Inbar used procedure 1 and a drawing of the construction to construct the parallelogram shown in Figure 1. In the second part she was asked to predict the DB of the figure she had just constructed. The analysis of Inbar's predictions reveals an unexpected view of order.

On the screen:

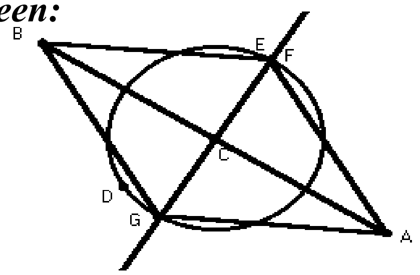


Figure 3: Inbar's construction

Varda: Can we drag point C? *(C is the midpoint of AB)*

Inbar: I don't think so. Dragging will move the whole figure together and nothing will change, neither the length nor the angles.

Eventually C is not draggable according to Inbar. For her dragging means a change in the relative locations or positions of the various components, a more inherent change in the figure than mere relocation.

Varda: Why?

Inbar: Because C is part of two lines and C depends on these two lines. I can't just move point C to here *(the cursor points on segment AB)* [and expect that] the whole figure will be re-arranged accordingly! I cannot move it.

Inbar correctly predicted that an intersection point can not be moved because it depends on two lines (she calls segment AB a line). But in her construction AB is the only parent of C. The other segment was constructed as a child of C (and other elements).

Varda: So can we drag point C?

Inbar: No. We can't. C depends on the whole circle. If you move point C, then it will not be the center of the circle. If you want to drag it, the whole figure will move and it will not be changed.

That made explicit the interpretation of order Inbar presented: C depends on the circle. The parent depends on its child.

Varda: Inbar, what will happen if we drag point E? *(E is a point on the circle).*

Inbar: Ah… ah… I think that it can be dragged because it is an arbitrary point… and it will move on the circle because it is an arbitrary point.

To clarify her perception of "a point on an object" Varda pointed the arrow to an exterior location:

Varda: Can you drag point E to this position? *(The cursor points to segment AB).*

Inbar: Yes. The whole figure will move too.

Varda: Why?

Inbar: Because point E is on it… It is on the drawing and if you move point E then the whole drawing will move depending on E.

Varda: What do you mean by the whole drawing? Would the circle move too?

Inbar: Yes! The point is on the circle.

Inbar expected to see the circle relocate according to the new location of point E.

Inbar's perception of the relationships between the elements of the objects is the opposite of the order of construction. It was surprising to find out that the same perception was revealed in other interviews as well, not only with students but also with teachers.

Is this an indication of tenuous mathematical knowledge? Or is it merely the 'wrong' genesis of the tool perception? Is there a tacit model that influences their expectations, and if so, is it software dependent? A briefly look at different software environments may help answer these questions.

DESIGNERS' INTERPRETATIONS OF ORDER OF CONSTRUCTION

Below is an example that suggests that DB depends on considerations other than purely geometric decisions; we call these design considerations. The example illustrates the construction shown in Procedure 2, Figure 1 within three DGEs: the Geometric Supposer, the Geometer's Sketchpad 3.10, and the Geometre Cabri II, by comparing the DB of one point (Figure 4, below).

The DB of the same procedure varies and not only in visual appearance. One of the diagonals of the parallelogram is the diameter of a circle. While dragging point A within the Cabri, the circle relocates without changing the radius. A similar DB occurs in the Supposer, but dragging point A in the Sketchpad changes the radius. Thus, different parallelograms are being created in each DGE. A possible explanation for the different DBs is different designers decisions regarding parent-child relationships.

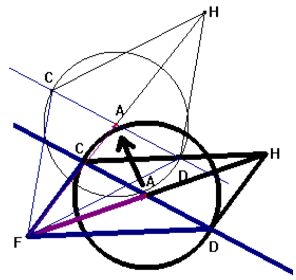
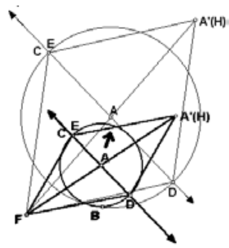
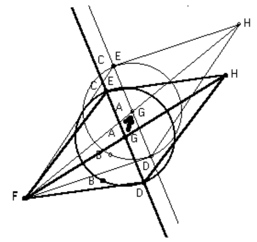
	Cabri II	**Sketchpad**	**Supposer**
Dragging point A, the center of the circle.			
Geometric family	Parallelograms having a diagonal of the same length	Parallelograms	Parallelograms having a diagonal of the same length

Figure 4: The DB of one parallelogram's vertex within three DGEs

The circle command in the Sketchpad and the Supposer produces a circle defined by two points: the center point A and point B on the perimeter. AB is the radius. The circle command in Cabri determines only the center A. In the Sketchpad points A and B are considered to be the parents of the circle, in the Cabri point A is considered to be the parent and the circle its child. In the Supposer, the three elements, the circle and points A and B, are considered as the same entity.

Dragging the center of the circle within the Cabri and the Supposer produces the same DB despite different parent-child relationships. The different DB of the Sketchpad can be interpreted resulting from the definition of both A and B as parents of the circle.

WHY DOES THIS MATTER?

Dynamic Geometry Environments (DGEs) have become standard tools for students, teachers, and mathematicians. Moreover, using DGE established new norms of learning geometry and deepen our understanding of tool-situated learning. Dragging geometric construction is in the foundation of these environments and unquestionably enhances the esthetics and power of DGEs. At the same time it changes the nature of the dependencies declared in the construction thus it introduces new complexity to the learning in DGE. It seems that Euclidean axioms and theorems are partial components that influence the DB of the construction.

To better understand the stages of instrumental genesis of DGE, software design decisions regarding order of constructions should be further investigated, guided by such questions as: What do users observe while dragging one element of the construction? How do they interpret the DB and what are their perceptions of it?

REFERENCES

Chazan, D., & Yerushalmy, M. (1998). Charting a course for secondary geometry. In R. Lehrer & D. Chazan (Eds), *Designing Learning Environments for Developing Understanding of Geometry and Space*. (pp. 67-90). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Goldenberg, E. P., & Cuoco, A. A. (1998). What is Dynamic Geometry? In R. Lehrer & D. Chazan (Eds), *Designing Learning Environments for Developing Understanding of Geometry and Space.* (pp. 351-368). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Healy, L. & Hoyles, C. (2001). Software tools for geometrical problem solving: potentials and pitfalls. *International Journal of Computers for Mathematical Learning*, 6, 235–256

Hölzl, R. (2001). Using Dynamic Geometry software to add contrast to geometric situations – A case study. *International Journal of Computers for Mathematical Learning.* 6, 63-86.

Jones, K. (2000). Providing a foundation for deductive reasoning: students' interpretations when using Dynamic Geometry software and their evolving mathematical explanations. *Educational Studies in Mathematics*, 44, 55-85.

Laborde, C. (1993). The computer as part of the learning environment: the case of geometry. In C. Keitek, & K. Ruthven (Eds.), *Learning from Computers: Mathematics Education and Technology.* (pp. 48-61) NATO ASI Series, 121.

Marioti, M. A. (2002). The influence of technological advances on students' mathematics learning. In L. D. English, (Ed.) *Handbook of International Research in Mathematics education.* (pp. 695-723) Hillsdale New Jersey: Lawrence Erlbaum Associates.

Parzysz, B. (1988). "Knowing" vs. "seeing": Problems of the plane representation of space geometry figures. *Educational Studies in Mathematics*, 19(1), 79-92

Straesser, R. (2001). Cabri-géomètre: Does Dynamic Geometry software (DGS) change geometry and its teaching and learning? *International Journal of Computers for Mathematical Learning.* 6, 319-333.