CERME9

Proceedings of the Ninth Congress of the European Society for Research in Mathematics Education

Editors: Konrad Krainer and Naďa Vondrová
Organized by: Charles University in Prague, Faculty of Education
Year: 2015

erme
european society for
research in
mathematics
education

Editors: Konrad Krainer, Naďa Vondrová

Publisher: Charles University in Prague, Faculty of Education and ERME

Place: Prague, Czech Republic

Year: 2015

ISBN: 978-80-7290-844-8

Recommended citation:

Author: Title of the contribution. In K. Krainer & N. Vondrová (Eds.) (2015). *Proceedings of the Ninth Conference of the European Society for Research in Mathematics Education (CERME9, 4-8 February 2015)* (pp. xx-yy). Prague, Czech Republic: Charles University in Prague, Faculty of Education and ERME.

Formalising functional dependencies: The potential of technology

Giorgos Psycharis

University of Athens, Department of Mathematics, Athens, Greece, gpsych@math.uoa.gr

This paper proposes "formalising functional dependencies" as an approach to address critical aspects of the potential of digital technologies for the teaching and learning of functions. This approach focuses on the role of the available tools in supporting students' transition from experiencing dependencies in terms of non-algebraic digital representations to expressing these dependencies formally. To illustrate the approach, data from two studies based on the use of two distinct computational systems are analysed. Key aspects of their potential include: work with dependencies at the level of magnitudes, specially designed functionalities and dynamic interplay between symbolic and non-symbolic representations of functions.

Keywords: Functions, dependency, formalism, digital technologies.

INTRODUCTION

The notion of function occupies a central position in school mathematics curricula but it constitutes a rather difficult topic for many students. Evidence identified by research concerns issues such as students' difficulties in understanding function as covariation and dealing with algebraic symbolism (Kieran, 2007). The development of new modes of representation within specially designed technological tools that allow considering functional dependencies through the use of non-standard representations (including non-algebraic ones) has generated further interest as regards their potential to deal with the above mentioned difficulties. One distinct feature of these tools is that they are designed to make the symbolic aspect of function more accessible and meaningful to students, especially through multiple linked representations including some sort of combination of visual or geometric representations (e.g., dynamic geometry) and algebraic multirepresentation, possibly including Computer Algebra Systems (CAS) or other symbolic forms (e.g., algebraic-like formalism) (Mackrell, 2011). The interactive and dynamic character of the corresponding digital representations have brought to the fore the need to acknowledge both the transformative potential of the corresponding technologies and the opportunities provided for meaning generation. This need was reinforced by the fact that such systems encourage different levels of interplay between symbolic and non-symbolic representations (e.g., direct manipulation of mathematical objects) and different kinds of algebraic expression that can be aligned or not with standard mathematical notation. In this study, I am particularly sensible to the possibility offered by particular computational environments to students to make sense of function through modelling dependencies in a non-algebraic/symbolic (e.g., geometrical, iconic) setting before passing to formal expressions of these dependencies and to mathematical functions. In this process, there is always a transition from experiencing dependencies through the use of non-algebraic/symbolic digital representations to recognising which of these dependencies constitute functional ones and expressing them formally using the available symbolic representational forms. This transition is far from trivial for students. Apart from the novelty of the used representations, well-known problems that students face with functions and algebra are also brought to the fore (Kynigos et al., 2010). Thus, the potential of the corresponding technologies needs to be addressed. This is the general goal of this paper. Based on the integrated framework developed by Lagrange and Psycharis (2014), I adopt a similar approach – I call it *formalising functional dependencies* – to address such potential.

FORMALISING FUNCTIONAL DEPENDENCIES

The approach formalising functional dependencies is developed around the need to address the following issues as regards the students' transition from non-symbolic/algebraic to symbolic/algebraic representations of function and the coordination between them in technology enhanced mathematics: (1) the role of the available tools in supporting different levels of students' work with dependencies (e.g., modelling, exploration), (2) the students' activity to 'translate' the modelled functional relations in symbolic language and to conceptualise the connections between different representations of function, and (3) critical aspects of the students' difficulties with functions (i.e., covariation, symbolism). Below, I present briefly the three parts of the theoretical work that underlie these issues and constitute the approach.

Levels of students' work with dependencies. Lagrange and Artigue (2009) developed a conceptual framework for functions and algebra in order to address students' work with dependencies. Taking an epistemological point of view, they situated students' activities for approaching functions at three levels: (1) activity in a physical system (e.g., dynamic geometry, a simulation): students can experience dependencies sensually in a physical system through observation of mutual variations of objects; (2) activity on magnitudes: the idea of function is linked to dependencies between magnitudes which is expected to support students' consideration of functions as models of physical dependencies; (3) activity on mathematical functions: students work with mathematical functions of one real variable, with formulas, graphs, tables and other possible algebraic representations.

Situated abstraction. Noss and Hoyles (1996) introduced the notion of situated abstraction to address abstraction within computational media as a meaning generation process in which mathematical meanings are expressed as invariant relationships, but yet remain tied up within the conceptual web of resources provided by the available computational tool. In this perspective, a 'situated abstraction' approach to students' conceptualisation of function within a particular computer-based setting involves meaning generation evident in the concretion of generalized relationships by students through the use of the available tools and structures.

Function as co-variation and the role of symbolism. The essence of a co-variation view of function is related to the understanding of the manner in which dependent and independent variables change as well as the coordination between these changes (Thompson, 1994). However, this dynamic conception of simultaneously variation between magnitudes is rather difficult for the students especially when mathematical symbolism is involved. Research has been showing rather conclusively that the idea of independent variable, the algebraic expression of functions and its connection to other representations constitute obstacles for many students even for those beginning to study more advanced mathematics (Kieran, 2007).

In order to concretize the approach, I consider here two computational systems, both dealing with functions through innovative representations and functionalities, but different in many other aspects. One is *eXpresser*, a microworld designed to support 11–14 year-old students in their reasoning and problem-solving of generalisation tasks (Noss et al., 2009). It provides an 'algebraic' language, which involves the use of numbers and variables, with the aim to support students to construct relationships between patterns. The other system is *Casyopée* (Lagrange, 2010). It offers a dynamic geometry window incorporating representations of measures and of their covariation connected to a symbolic environment designed to support students' work on mathematical functions. Both systems offer opportunities for students to understand key actions in the process of modelling a dependency into a functional relation. However, eXpresser uses non-standard 'symbolic' representations while Casyopée's symbolic forms are consistent with current notations at secondary level. Next, I adopt the approach formalising functional dependencies to analyse data of eXpresser's and Casyopée's use in two respective studies so as to make sense of their potentialities for functional meaning making.

FUNCTIONAL RELATIONS IN FIGURAL PATTERN TASKS

eXpresser. The microworld affords the creation of coloured patterns in the construction area (Figure 1) by repeating a building block of several tiles ('unit of repeat'). The students can select tiles of different colours to construct the unit of repeat and then to define a pattern by specifying the number of repetitions and the appropriate number of coloured tiles in its property window (e.g., Figure 4). The number of coloured tiles can be represented iconically through expressions involving numbers that appear tied in a grey frame and 'unlocked numbers' – i.e. variables – and appear tied in a pink frame. A variable can be defined through a pop-up menu by 'unlocking' a particular number corresponding to an attribute of a construction (e.g., the number of red tiles, the number of repetitions) and provides a representation of an independent variable and its current value. Variables can be copied, deleted or used in operations (e.g., addition). Thus, through the use of variables, students can create relationships between two patterns of different colours based on dependencies (e.g., between the numbers of tiles of different colours). A pattern is shown dynamically (i.e., animated) by pressing the button Play (Figure 1). Then, the microworld picks random values for every variable and the model is shown dynamically in the construction area. Thus, students have the opportunity to see how their construction would look if the values of the independent variables of their current construction had different values.

It is important to mention that a model is always coloured only if the same independent variable has been used to build appropriate general expressions representing the total number of tiles for each one of the patterns used for the construction of the model. In different cases, the unfolded model/pattern appears to be distorted ('messed-up') and it is not coloured (e.g., see Figure 4b, c). Another feature of eXpresser is that of 'General Model' window (Figure 1): when the students animate a pattern this window shows different instances of the construction for different values of the various parameters in relation to the values assigned by the system in the representation appearing in the construction area. In order to colour their pattern in the 'General Model', the students have to build a general expression (i.e., the *Model Rule*, Figure 1) that always gives the total number of all tiles in the model (i.e., not just any pattern).

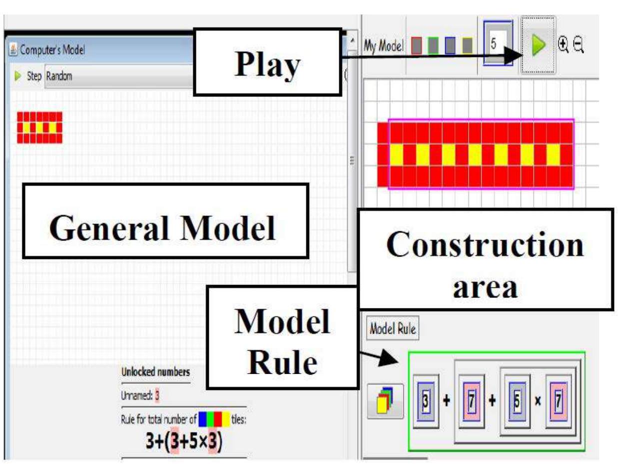

Figure 1: A pattern in eXpresser

The experiment. In the study with eXpresser (Zoupa, 2013), three case study groups of 13-year-old students (6 sessions for each group) were asked to construct and validate patterns through general expressions that underpin them. Since the students had not had any experience with patterns in their school lessons, the aim was to investigate if and how the microworld could help them to understand dependencies and express them using the system's structures and symbolic language. The data consisted of screen capture software files, files of students' work and video recordings. The data was analysed under a data grounded approach. Through the analysis of students' interaction with the available tools, episodes were selected to highlight the evolution of meaning generation for function.

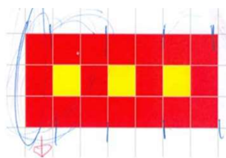

Figure 2

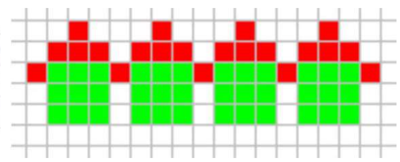

Figure 3

After an initial familiarization with eXpresser, the students were engaged in constructing the patterns shown in Figure 2 and Figure 3 consecutively. They had to allocate the correct number of tiles of each colour that were needed for the construction and then to create appropriate general relationships by using the same independent variable in the task. Next, I describe one group of students' work in four phases and corresponding steps that took place in the second and the third session.

Phase 1: Exploring the role of numbers and variables

Task: Construction of the pattern in Figure 2. (a) Constructing two building blocks (patterns): the first one constituted by the first column (3 red tiles) and the second one constituted by the second and the third column (6 tiles: 5 red and 1 yellow). (b) Considering the first pattern as specific. (c) Constructing the second pattern specifically for three repetitions (Fig 4a). (d) Unlocking the number of repetitions in the second pattern but keeping constant the numbers of red and yellow tiles. Feedback showing the construction was not coloured for different numbers of repetition in the construction area (Figure 4b). (e) Unlocking the number of red and yellow tiles without linking these (new) variables to the variable defined in the previous

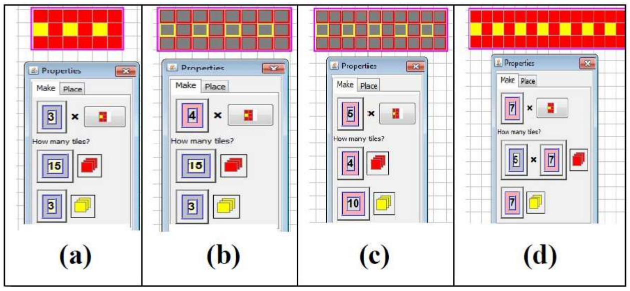

Figure 4: The properties window and the visual outcome (second pattern)

step. Feedback showing that the construction was not coloured in the construction area (Figure 4c).

Phase 2: Building functional expressions within a pattern

Task: Construction of the pattern in Figure 2. (a) Recognising the number of repetitions of the second pattern as an independent variable in the task. (b) Using this variable to express the number or red and yellow tiles (i.e., the number of red tiles is the same as the number of repetitions of the second pattern, while the number of yellow tiles is five times the number of repetitions of the second pattern). Inserting these expressions in the properties window through the choice 'replace' (Figure 4d). (c) Animating the model dynamically in the construction area. Feedback confirming the correct animation of it (Figure 4d).

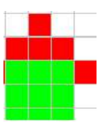

Figure 5a

Figure 5b

Phase 3: Building functional expressions between patterns

Task: Construction of the pattern in Figure 3. (a) Constructing three building blocks (patterns): the first one constituted by the one red tile (i.e. the red tile on the left part of the first house), the second one for the roof with 5 red tiles and the third one for the green square with 9 green tiles (Figure 5a). (b) Considering the first pattern as specific and constructing the second and the third ones as general by unlocking the numbers of repetitions and thus creating one independent variable in each one of them. (c) Running the models dynamically. Feedback showing that the construction was 'messed-up' due to fact that the two variables representing the number of repetitions in each pattern changed according to different (randomly chosen) values (Figure 5b). Recognising that the two variables had to take the same value. (d) Linking the two patterns by replacing the one independent variable with the other through dragging and the choice 'replace'. (e) Building appropriate functional expressions for the numbers of green and red tiles in the two patterns.

Phase 4: Expressing the general rule of the total number of tiles in the model

Task: Construction of the pattern in Figure 3. (a) Constructing a general expression giving the total number of all tiles (i.e. not just any pattern) in the Model Rule window through the use of the only independent variable (Figure 6 shows an immediate instantiation of this expression for three repetitions of the model). (b) Expressing the general rule through traditional algebraic notation with paper-and-pencil (i.e., 5x+9x+1).

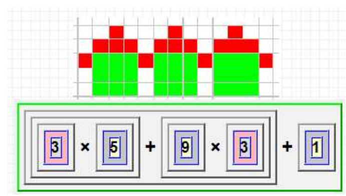

Figure 6

As regards the levels of dependencies, the physical system in eXpresser involves the dynamic reproduction of patterns. At the level of magnitudes, numbers of tiles and numbers of repetitions are involved. These magnitudes are concretized in the system as numbers in grey frames and variables in pink frames showing current instances of their values. I note that the dynamic change of the values (assigned automatically at random) to one variable is shown inside the pink frame of a variable when the corresponding pattern unfolds dynamically in the construction area. Thus, measures are 'encapsulated' within the corresponding magnitudes. As regards the notion of function, the independent variable is the number of repetitions of a building block created by the students while the dependent variable is the total number of tiles (of each colour). In this linear function, the input is the number of repetitions and the output the animated model. In all phases, while exploring the role of numbers and variables and experimenting with building different symbolic forms of general relations, the students considered together the physical system and the dependency between magnitudes. Thus, they worked with dependency and co-variation together at the level of magnitudes and at the level of magnitudes represented through variables.

From a situated abstraction point of view, the domains of students' meaning generation here involve: (a) making sense of the structure of the requested models in terms of specific and general patterns, (b) conceptualising the construction of appropriate building blocks, (c) identifying the independent and the dependent variables, and (d) conceptualising the formulation of functional relations between these variables. Building appropriate functional relations indicated the students' transition to the world of functions as it is embedded in the structures of the system.

As regards students' conceptualisation of covariation and symbolism, the role of feedback was critical. eXpresser provided a dynamic representation of covariation: animating the pattern had the effect of the construction dynamically changing as the values of the respective parameters changed automatically. In this process, 'messing-up' and 'correct colouring' challenged students to create and undertake changes in the symbolic form of the corresponding relations and at the same time to progress in their conceptualisation of the involved covariations as functions.

LINKING GEOMETRICAL DEPENDENCIES AND FUNCTIONS

Casyopée deals with various representations of functions consistent with school mathematics and curriculum. It provides a symbolic window with three registers: numeric, graphic and symbolic (Figure 8). Casyopée also includes a dynamic geometry window linked to the symbolic window. The geometric window allows defining independent magnitudes (related to free points) and also dependent ones (i.e., through the use of the "geometric calculation" functionality, see Figure 7 on the right) involving distances (e.g., lengths), *x*-coordinates or *y*-coordinates. The two windows are interconnected: objects defined in one window can be used in the other. Couples of magnitudes that are in functional dependency can be exported to the symbolic window and the system automatically can define a function. This function can be further treated by the students with all the available tools. This functionality –called "automatic modelling"– is expected to help students in modelling dependencies.

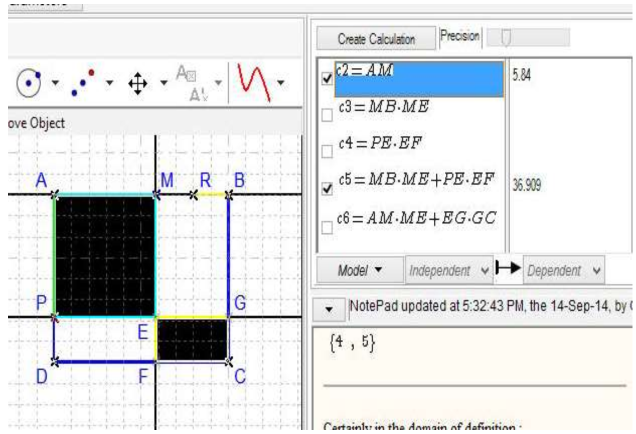

Figure 7

The experiment. In the study with Casyopée (Kafetzopoulos, 2014), three case study groups of 17 year-old students (six sessions for each group) were engaged in solving optimization problems through modelling geometrical dependencies. The aim of the study was to investigate how a computational medium linking CAS and dynamic geometry could help students make sense of function as covariation through: (a) conceptualising dependencies in geometrical situations and (b) modelling them as functions in order to solve the given tasks. Here I will report on students' work in the following task: "The owner of a rectangular estate *ABCD* (*AB*=10m, *AD*=8m) wants to design two gardens and two buildings inside it. In the given geometrical figure (Figure 7), we consider *A*(0;8), *B*(10;8), *C*(10;0), *D*(0;0). *M* is a point on *AR* (*AR*=8m) and *P* a point on *AD* so as *AMEP* to be a square. The figure *EGCF* is a rectangle. The gardens will cover the shaded part of the figure and the two buildings the rest of it. By moving *M* between *A* and *R*, different shapes of the gardens are designed. (1) Is there a position of *M* in *AR* for which the two gardens have the same area? (2) Is there a position of *M* in which the two gardens will have the same area as the two buildings? Justify by: (a) exploring the dynamic figure, (b) using the software to create functions modeling these questions, and (c) using the available tools to find the solutions". Since the main focus of the task was on students' conceptualization of function as covariation, the figure was already prepared for them. The students had to: (a) make sense of *M* as the only free point in the figure, (b) recognize *AM* as an independent variable and use it to define functions of areas, (c) work with different representations of functions. Next, I describe the work of one group of students in four phases and corresponding steps that took place in the last two sessions. The data and method of analysis were similar to the ones described in the experiment with eXpresser.

Phase 1: Exploring dependencies in the geometrical model. (a) Experimenting with the dynamic aspects of the figure by dragging points. (b) Recognising *M* as the only free point in the model. (b) Defining measures for different magnitudes (e.g., the areas of *AMEP* and *EGCF* and their sum) as geometrical calculations (Figure 7, right). (c) Observing covariation at a perceptual level (i.e. how dragging *M* changes the shapes of the shaded parts) and numerically (i.e. through the changes in the values of the corresponding magnitudes). *Phase 2: Identifying independent and dependent variables.* (a) Choosing an independent variable after recognizing dependencies between co-varying magnitudes (e.g., *AM* for the area of *AMEP*). (b) Using the same independent variable for the area of *EGCF* after checking that dragging *M* changes the area *EGCF*. (c) Choosing *AM* as an independent variable for the sum of areas (i.e. *AMEP*+*EGCF*, *MBGE*+*PEFD*). *Phase 3: Working with algebraic functions through automatic modelling.* (a) Exporting functions to obtain formulas for four functions (two for question 1 and two for question 2, Figure 8, left) through automatic modeling. (b) Working further on the algebraic functions to solve the problems, i.e. making equal two functions and solve. Identifying the position of M in question 1 (AM=40/9) and question 2 (AM=4 or 5). *Phase 4: Linking different representations of algebraic functions.* (a) Interpreting the answers to questions 1 and 2 given through the equality of functions by coordinating different representations, e.g., linking table, geometry and graphics, focusing on the common point of the two graphs, changing the step in the corresponding tables for more precise values (Figure 9). (b) Conceptualizing the addition of two already defined functions as a new function (e.g., *AM*→*AM*•*ME*+*EG*•*GC*).

As regards the levels of dependencies, the physical system in Casyopée is the dynamic geometry window which provides the context for modelling the problem in the geometrical setting and opportunities for creating and animating geometrical objects. At the level of magnitudes, the students can use the geometric calculation functionality to construct magnitudes in the form of symbolic objects that can be dependent to the geometrical situation (e.g., expressions of areas). These magnitudes have a dual status in the system since they are concretized symbolically as parameters (*c0*, *c1*, etc.) and numerically as measures whose values can change dynamically (e.g., through dynamic manipulation of the dependent geometrical objects). Thus, the students can be engaged in exploring covariation of pairs of magnitudes, modeling functional dependencies algebraically (through automatic modeling) and working further with mathematical functions. Thus, the combined use of geometric calculation and automatic modeling supports students' transition from the world of measures to the world of mathematical functions through work with magnitudes. This transition is evident in students' activities described above: work in the physical system (phase 1a) was followed by work with magnitudes (e.g., definition of geometric calculations for areas) and observation of covariations (phases 1b, 1c), and then further extended to include identification of independent variable (phase 2), definition of functions through automatic modeling (phase 3) and problem solving by linking different representations of functions (phase 4).

From a situated abstraction perspective, the layers of meanings for function here involve: (a) making

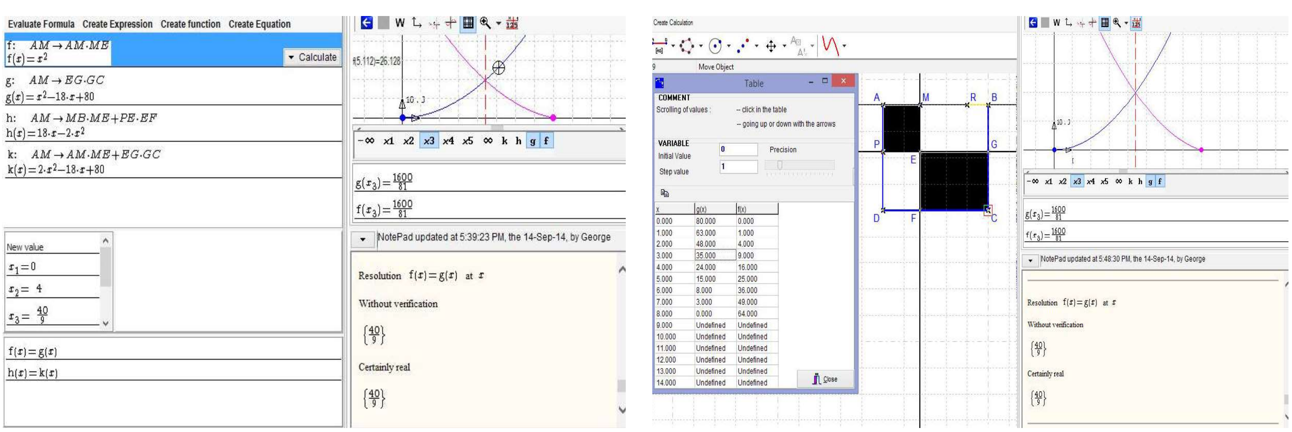

Figure 9

sense of the dependency between M and the areas influenced by its move, (b) conceptualizing the creation of relevant geometric calculations, (c) conceptualizing the idea of independent variable representing a geometrical object and using it to express functional dependencies through automatic modelling, (d) abstracting these dependencies at the algebraic level and use the corresponding functions to solve the requested problems.

As regards the symbolic aspect of function, the formulas taken in Casyopée seemed to have legitimized the use of all the available representations of functions by the students. Although the students were able to explain the provided formulas of functions, they chose to work with these formulas at an operational level for solving the given tasks (i.e. through the equality of functions). In doing so, they were engaged in linking different representations of functions.

CONCLUSION

As regards the levels of dependencies, the analysis revealed that eXpresser and Casyopée favour students' work with dependencies at the level of magnitudes as a critical part of their passage to formalisation. In particular, the analysis indicated the importance of working with magnitudes as a bridge between sensual experience of dependencies and symbolic expression of functional relations. In eXpresser, dependencies between magnitudes and visual representation of their covariation (i.e., through dynamic reproduction of patterns for random values) seemed to support the students' articulation of general relations and favour a structural understanding of patterns. Dependencies of this kind in Casyopée seemed to facilitate the students' transition from dynamic geometry and the world of measures to the world of mathematical functions. The analysis of the students' activity according to situated abstraction helped us to capture the progression of their conceptualization of functional dependencies taking into account the role of particular functionalities (e.g., automatic modelling) and feedback (e.g., messing-up). Experiencing covariation through dynamic interplay between symbolic and non-symbolic representations of functions helped the students to make sense of the role of independent variable and symbolism in expressing functional relations and connecting the formalism embedded in the tools to the formalism of school mathematics. This is an indication that the two systems can be used to address well-known and researched difficulties of students with algebra and functions (e.g., recognising independent variable, articulating functional relations and expressing them symbolically) and promote a meaningful transition to algebraic thinking. Thus, the potential of the two systems can be recognised in the direction of enriching representations of functions with new non-symbolic and symbolic ones and of enlarging students' possibilities to construct functional meaning by making connections between these representations.

REFERENCES

Kafetzopoulos, G. I. (2014). *Conceptualisation of function as covariation through the use of the computer environment Casyopée* (Master's thesis). University of Athens, Greece.

Kieran, C. (2007). Learning and teaching algebra at the middle school through college levels. In F. K. Lester Jr. (Ed.), *Second handbook of research on mathematics teaching and learning* (pp. 707–762). Charlotte, NC: Information Age Publishing; Reston, VA: NCTM.

Kynigos, C., Psycharis, G., & Moustaki, F. (2010). Meanings generated while using algebraic-like formalism to construct and control animated models. *International Journal for Technology in Mathematics Education, 17*(1), 17–32.

Lagrange, J.-B., & Artigue, M. (2009). Students' activities about functions at upper secondary level: a grid for designing a digital environment and analysing uses. In M. Tzekaki et al. (Eds.), *Proceedings of the 33rd PME Conference* (Vol. 3, pp. 465–472). Thessaloniki, Greece: PME.

Lagrange, J.-B., & Psycharis, G. (2014). Investigating the potential of computer environments for the teaching and learning of functions: a double analysis from two research traditions. *Technology, Knowledge and Learning*, 19(3), 255–286.

Lagrange, J.-B., (2010). Teaching and learning about functions at upper secondary level: designing and experimenting the software environment Casyopée. *International Journal of Mathematical Education in Science and Technology, 41*(2), 243–255.

Mackrell, K. (2011). Integrating number, algebra, and geometry with interactive geometry software. In M. Pytlak et al. (Eds.), *Proceedings of the 7th CERME* (pp. 691–700). Rzeszów, Poland: University of Rzeszów.

Noss, R., & Hoyles. C. (1996). *Windows on mathematical meanings*. Dordrecht, The Netherlands: Kluwer.

Noss, R., Hoyles, C., Mavrikis, M., Geraniou, E., Gutierrez-Santos, S., & Pearce, D. (2009). Broadening the sense of 'dynamic': a microworld to support students' generalisation. *ZDM, 41*, 493–503.

Thompson, P. W. (1994). Students, functions, and the undergraduate curriculum. In E. Dubinsky, A. H. Schoenfeld, & J. J. Kaput (Eds.), *Research in collegiate mathematics education I* (Vol. 4, pp. 21–44). Providence, RI: AMS.

Zoupa, A. (2013). *Construction of meanings for generalisation through figural pattern tasks* (Master's thesis). University of Athens, Greece.