Generating cultures for mathematical microworld development in a multi-organizational context

Chronis Kynigos

Educational Technology Lab,

University of Athens

School of Philosophy,

Faculty of Philosophy, Education and Psychology,

Dept. of Education.

and Computer Technology Institute.

Kynigos@cti.gr

Abstract

This paper discusses methodological issues of mathematical microworld development integrated with generating innovation in the school setting. This is done by means of vignettes of key episodes in our eight year-long experience of developing a component architecture for educational software based on Logo as a scripting language. The vignettes touch on the problems of collaboration between organizations and people of different expertise. They also address issues to do with the school and the classroom as social systems, with the method for implementing innovation and with curriculum design, teaching and learning. A set of issues which emerged to be problematic are outlined and discussed; the different priority systems involved, the amount of investment in collaboration, the differing discourses and epistemologies, the notion of a product, the interdependencies and the contrast between reform and innovation versus instant fit. It is suggested that awareness needs to be raised as well as methods for dealing with these factors in order to generate cultures developing and using exploratory software.

Introduction

This article discusses emerging organizational contexts within which mathematical microworlds and corresponding pedagogies may find new grounds to develop. I argue that besides developments in the fields of technological and educational research, the development of these microworlds and pedagogies is influenced by the ways in which a much larger variety of organizations and communities are progressively participating in the shaping of perspectives related to the use of computational and communication technologies for education and the ways in which these organizations influence corresponding educational reform. The discussion attempts to identify some issues that may be crucial to the role of the mathematical microworld community in this wider setting of technological infusion in educational systems. This is done by means of a set of illuminative vignettes, grounded in a longitudinal eight year - long experience of infusing a mathematical – constructionist perspective into a series of consecutive projects. These were all funded by the European Community, which explicitly supports collaborations between a diversity of institutions and educational systems. They shared as central objective to generate communities of people collaborating in developing, reflecting on and proliferating the use of exploratory software. They involved the on – going development of E-slate[1], a toolkit for developing a wide range of exploratory software, the design, management and support of a large – scale systemic initiative to integrate technology into the Greek education system and classroom - based research on teaching and learning processes emerging from implementing innovative programs based on the use of E-slate software.

Emerging organizational contexts

The process of developing exploratory software for mathematics requires a synthesis of expertise: in computer science for software development; in mathematical teaching and learning theory; in the method of integration and support of the use of the software in real school settings, and in the production of polished software and respective materials to the level of professional-looking products. These types of expertise are typically found in different organizations and departments, each with its own history of evolution and agenda for progress. In the past twenty years, however, the development of such software was funded mainly as focused research and was based in a university, research institute and occasionally, a company typically with some state or federal funding and tight connections to academia. To deal with the need for composite expertise, these organizations would most often employ *complementary actors*, e.g. either a programmer or an educationalist, depending on the expertise missing at the respective department. In many cases, these people would become *hybrid* actors, in the sense that they would acquire enough complementary understanding so as to effectively act as part of a team based on alien expertise. It was through this type of uni – organizational setting that the community working on the conception of technologies supporting the generation of mathematical meanings and the development of our understanding of this process as essential for learning mathematics was developed[2].

In the early eighties, however, this kind of setting and this kind of perspective in using technology for learning was the norm. Mathematical microworlds were one of the very few computational environments designed for learning. In these computational environments, getting a computer to do things involved a very neat way of making use of the available technology. It enabled the learner to express ideas in symbolic (mathematical) form by means of text and when appropriate, to see the results represented in function-derived graphics (as in the turtle graphics part of the Logo language). In itself, constructing and exploring through executable ideas, and reflecting on these through computer feedback and editing is an important mechanism for microworld functionality. The creation of focused and coherent environments along with a set of tools for making constructions or means for exploring (to paraphrase Clements and Sarama's definition of microworlds in this volume) is to a significant extent possible with these technological means alone.

Furthermore, symbolic expression and construction by means of mathematical formalism was not only seen as an important part of such software, but also as central to the nature of mathematical learning (Harel et. al., 1991, Noss and Hoyles, 1992, Kynigos et al., 1997). The constructionist - exploratory (Kafai and Resnick, 1996, diSessa, et. al., 1995) perspectives for learning with technology constituted significant advances to our understanding of the mathematical learning process. Applied research in implementing or attempting to institutionalize educational practices based on these, such as Noss, 1985, Hoyles and Sutherland, 1989, were amongst the first experiences of using computational technology in the school system. A central idea stemming from this community is the distinction between a computational tool and an instrument (Lagrange et. al, 2001): a tool is a computational artifact, while an instrument refers to how that artifact was mentally constructed by the user, that is, how the user conceives its constraints and possibilities. Another, is the process of integrating software development with the study of educational processes emerging from its use - where a single piece of software may become a set of different instruments - and finally with the institutionalization of its use in the educational setting (Papert, 1993, Hoyles, 1993).

Since then, however, the advent of new interfaces, media, technological power and speed provided the possibility of extending mathematical microworlds' functionalities in a variety of ways, as for instance, using direct manipulation in dynamic geometry environments, multimedia authoring, realistic simulations, data handling and the use of a series of computational ideas such as parallel or object oriented programming. These developments have given rise to a wider variety of representations of mathematical ideas and objects and the means for manipulating or operating on these (see Edwards, 1998 for a discussion on the notion of microworlds as representations). Such environments have also provided the possibility to highlight the mathematical aspects of a wider set of situations, such as simulations (see SimCalc, end of text).

In parallel, these new technologies together with the explosive developments in communications technologies have been used to develop a large variety of technological environments *outside* mathematics, respectively embodying a breadth of perspectives on learning across a variety of subject areas and themes. Experiments or explorations with simulations (Interactive Physics), with the use of large and well organized repositories of information (Perseus), with data handling (Tabletop) and through participation in ‘knowledge building’ debates (CSILE) are some key student activities made possible with the new types of software available. The development of this kind and span of software has played an important role in bringing a much larger number of communities into the game, and very often perspectives of learning within these communities differ widely.

For instance, there are communities *outside* educational research taking an interest in integrating the use of communication and computer technology in the educational system: the computer applications branch of the R&D community, the software development industry, the telecommunications agencies and organizations and the educational policy makers (often assuming the role of ‘reformists’). These communities bring alternative perspectives on the essence of and the method for developing and using software in education.

Moreover, in some parts of the world, such as the European Community (E.E.C.), there is a consistent and explicit encouragement for these perspectives to integrate with each other in the last five years or so. In the case of the E.E.C., the agenda for this is *not* educational in the narrow sense, but more socio-political and developmental in nature: to create intercultural sensitivity and to generate more synergy between academia and industry. Researchers are thus faced with explicit encouragement to engage in R&D activity participating in multi-organizational consortia, rather than working through their organization and hiring hybrid actors. Not only this, but the directive is that these organizations should be a mix of academic, private sector and systemic (schools, local authority and ministry departments) institutions and that they should be situated in a variety of countries.

The interest shown by new communities, coupled with the encouragement of multi-organization projects, provide new potential and at the same time new complexity and challenge to the constructionist – exploratory community. Given that organizations incorporate sustained and institutionalized expertise, there is the potential to have unprecedented support in all aspects of the development, production and infusion of exploratory software use in educational settings. There is the challenge, however, that the educational purpose of exploratory software has to be communicated and integrated with powerful communities with different perspectives and priority systems with respect to education. The traditional processes and methods of developing exploratory software for education seem very idiosyncratic when seen through perspectives held in the wider communities involved in educational reform and in software development in general. At the same time, however, it is this community that has the expertise and deep understanding of integrated software development for educational change.

In this paper, I attempt to synthesize some key problematic issues which may face this community, in situations of collaboration with different organizations. These issues will be drawn from four vignettes describing different facets of an eight year - long experience of an education research team, a developers' team and a small number of schools working together with other organizations to develop exploratory software and integrate it in real educational practice. The approach to identifying these issues is generative and emergent. As argued above, rather than adopting a theoretical perspective deriving from a single discipline, I take the more ecological stance of 'naïve observer' (Goetz and Lecompte, 1984) which is at the heart of the qualitative research paradigm. The objective is to contribute to the illumination of the problem and to learn how to ask more focused questions regarding the potential and pitfalls of the role of the constructionist community in multi – organizational contexts.

Perspectives on Educational Software Development

A first way to approach the problem is to consider the ways in which some specific communities recently engaged in projects concerning technologies for education might perceive the methods and principles of the constructionist community inherent to the design of educational software. Here are some of the most central ones which may seem idiosyncratic to the outside world:

- the software is meant to facilitate innovation and therefore some change in activity, attitude, perceptions and understandings in both teachers and pupils;
- the same piece of software may be used by different people for different activities at various levels of sophistication;
- the software is designed to be used primarily for knowledge generating activity with some personal meaning and not for following directions, gathering or observing information, answering questions or simply observing things which are going on;
- in many cases, users will construct things with the software (not just observe or test out);
- what users do with the software may well be a surprise to the original designers;
- software development needs to be integrated with use in contexts that are as realistic as possible;

- there is a tendency for more emphasis on the context within which the software is used and on the activities rather than on the actual development of the software itself;
- there is a lot of effort spent on building an understanding between actors with different expertise, perspectives and stakes in the development process;
- the roles of developing, testing authoring and using are purposefully not clearly defined (see also diSessa, 1997).

This approach to software development is not necessarily understood, respected or supported by organizations and people outside this culture. Here are some viewpoints held in organizations and communities involved in education, but not within the microworld development culture.

From a systemic point of view, the school and the educational system seem to fight back against innovation (Papert, 1993), or to transform an innovation so that it finally fits in traditional practice (Hoyles, 1993). Its thus actually harder for software designed for innovation to be immediately used and understood. Software designed to fit a traditional classroom paradigm has much more chance of instant success.

From the point of view of educational reformers, i.e. educational policy makers advocating systemic *change*, a commonly held perspective is that of *organizational* reform, where the emphasis is on total quality management. This means that the key feature of operation is to make a very clear and detailed plan of processes and products and then to evaluate the process by means of testing the extent to which they were met. They key feature is to control processes and activities. In an eloquent argumentation in terms of the discourse used in organization reform, Prawat suggests that this commonly adopted “expect and inspect” approach is unsuitable for educational reform where a “learning community” approach allowing personal investment, messy and arduous procedures and emergent development is needed (Prawat, 1996). The development of exploratory software is an untidy and unpredictable process which grows together with the learning community engaged in integrating the software in classroom practice. To the T.Q.M. reformist, this could look like an loosely planned, expensive and long term investment with uncertain return.

From a commercial perspective there is focus on users' actual needs which are easily digestible and widely understood so that there can be immediate use of the software. The tendency here is to create the continual consumer (Noss, 1995), by means of creating widespread need to buy new versions of a piece of software, each with a number of new features (Eisenberg, 1995, Solloway, 1998). This is not always in line with the notion of the *continual user*, who buys a support service rather than a line of products. In these terms, if the objective is for exploratory software to be used, a continual "support service" is needed so that change in educational practice may occur.

Furthermore, there is growing interest in education by organizations contributing to wider issues of infrastructure, such as telecommunications, where the emphasis is on equal distribution and on delivery (Noss, 1992) in large scale projects. The main focus there is on information flow and information management. Knowledge is seen as equivalent to information and learning as equivalent to information gathering and processing. Activity, expression, construction, experimentation, generation of new meaning which are all central to the use of exploratory software, are outside this frame of reference.

Finally, the emphasis is often on the programming aspect (code) of the development of the software rather than on the creation of cultures of development and use in real contexts (diSessa, 1997).

Microworlds and exploratory software in general, have been seen as a vehicle for innovation – even though in practice the innovation has often been transformed to fit the system. In many cases they have been seen as a tool for schools. Also, in many cases, such environments have been developed by means of a method integrating development and school use. Educational software in general requires a very high level of expertise and a lengthy and costly development process and microworlds are at the high end of this.

The context

History

Since 1993, we have been involved in the integrated development of exploratory software, through a series of projects, each involving a different consortium of partner organizations. Initially, funding came from sources interested in supporting R&D projects in Europe[3]. Some success in these brought on our involvement in developmental projects, supporting the Ministry of Education's policy for the integration of New Technologies in secondary education[4].

For the purposes of this paper, we will use vignettes from the collaboration between two groups of people in two organizations based in Greece, which has been active continuously in all the above projects. One group, based at the Computer Technology Institute in Patra, has played the role of designing the technical specifications, and developing E-slate, an authoring environment for developing exploratory software. The main idea is that E-slate provides a set of pieces of generic software as building blocks for 'non – technical' users to put together in different configurations to create their own pieces of software.

This architecture and its specific use for educational software was – and still is – an object of R&D work (Roschelle et. al., in press). It provides a very high level of interoperability and reusability of components, enabling developers to build on components and functionalities instead of having to start from scratch for each piece of software. It also enables educationalists to make their own configurations of interoperating components and to create their own software and functionality. This can be done either with a direct manipulation interface or by means of a scripting language with which components can be connected and their behaviors defined and controlled. A crucial aspect of the collaboration of these two teams is that we have chosen a scripting language embodying mathematics and encouraging mathematical expression, i.e. Logo (Kynigos et. al, 1997).

The other group, based at the Education Department, School of Philosophy, University of Athens, took the role of designing functional specifications, microworlds and activities, educating teachers and carrying out educational research in a number of school sites. Each project involved the development of one or more pieces of software focusing on a particular age group and subject matter. The functionality of each piece of software attracted the research interest of both groups, since the component architecture invited a reappraisal of granularity and connectivity of the software components (Kynigos et. al., 1997). In each project, the software was used in five schools both as a means to provide feedback for further development and as a vehicle for carrying out research on a series of aspects of integrating innovation in the school setting. Collaboration with schools has been systematic and long term rather than the restricted implementation of short teaching experiments.

Activities and roles

Although this collaboration began with emphasis on the notion of integrated (rather than fragmented) development, the breakdown of work resulted in the emergence of the following activities and corresponding roles:

- component architecture design and development
- software design and development
- activity design and development
- collaboration with schools and school support
- teacher education

- technical infrastructure
- research involving classroom observation, tests and interviews

This became the arena for collaboration in the sense that although each person involved adopted one or a small subset of these roles, they needed to understand the activities of people in other roles enough so that the whole system of development would progress in a coherent way.

The software

E-slate is an authoring system for developing exploratory software of a wide variety of subjects, functionalities, targeted age groups and levels of use. Its authoring features are designed to allow 'deep structure access' (Di Sessa, 2000), i.e. rather than simply inserting content and defining its form and sequence, teachers are able to construct structures and functionalities. A core characteristic of the software environments developed with this rationale is their learnability, their all – embracing metaphors and their transparency with respect to the computer (as little "magic", or "black boxes" as possible). To a certain extent, e-slate purposefully makes compromises on these in order to provide teachers and students with ready - made higher - level and technically efficient building blocks, which we call 'components'. We have only recently begun to investigate the ways in which teachers and students learn to create software with e-slate[5], whether it provides enough user access to its functionalities and structure and the extent to which software constructions are interesting and original. Two features of e-slate are important. The available building blocks, i.e. generic pieces of software called components and the authoring metaphors, i.e. plugs and scripts. There are five categories of components:

- Information Handling (database, query, etc.),

- Simulation support (physics simulator, map, etc),
- User Interface controls (buttons, sliders, etc),
- Visualization (graphers, Venn diagrams, etc),
- Symbolic expression (Logo language) and
- Media handling (TV, sound, etc).

The rationale for having these components is to meet teachers and students half – way, that is, to provide them with generic pieces of software designed so that each can be used in many different configurations and roles (Kynigos 2001). Some of these are technically quite complex – the design rationale was that granularity would be decided on the basis of the potential for each component to seed creative ideas for its use in many different microworlds.

There are two metaphors for connecting them to construct “microworlds”, a word we use to signify E-slate creative component configurations and functionalities (for a discussion of the term, see Edwards, 1998). The plug metaphor allows the making of pre-fabricated connections by means of an icon-driven interface (the plugs). The scripting metaphor allows user defined connections by means of a programming language (Logo). A fundamental part of using E-slate is to create component combinations by connecting them together and building specific tools and behaviors. This can be done by the “connecting plugs” metaphor and through Logo, extended so that each component carries its own connectivity primitives. E-slate is thus programmable, tweakable and pokable (to use diSessa’s terms, 1997), but from the level of ready made components and upwards.

The research

Research has focused on a variety of aspects of classroom practice within the context of generating innovation with the use of e-slate. Such aspects have been:

- pupil collaboration (e.g. Kynigos and Theodosopoulou, in press)
- concept specific (e.g. Kynigos and Psicharis on the notion of curvature, 2001, Kynigos and Giannoutsou on the notion of spatial awareness and orientation, 2001)
- teacher beliefs and practices (e.g. Kynigos and Argiris, 1999).

In this paper, we take a particular action line in these projects, the development of e-slate software for mathematics, the design of respective microworlds, school implementation and research on different aspects of educational practice. We focus on a mathematical component which we called “the variation tool”, on a Logo component which extends traditional Logo to the role of a scripting language and on a database component (Kynigos et. al., 1997). The variation tool is designed so that it provides a kinesthetic means for continually changing the independent variable of the respective world to which it is connected and observing what remains constant and what changes. In this case, when the language, turtle, canvas and variation components are connected to each other, execution of a variable procedure with any value for the variable(s) and clicking on the turtle’s trace “energizes” the variation tool which recognizes which command resulted in that particular trace (fig. 1). A slider appears for each variable with editable range and step. Dragging the slider results in a continual reshaping of the figure according to the corresponding variable value. The effect is that of the same figure dynamically changing form (in a way similar to that of Geometry Sketchpad). More important, it gives a feeling of the way things change and the rate of change. A first example is that in figure 1, where the procedure ‘parallelogram’ was executed with values 50, 50 and 90 to construct a square. The variation tool was then activated and the latter two sliders dragged to form the parallelogram shape in the figure.

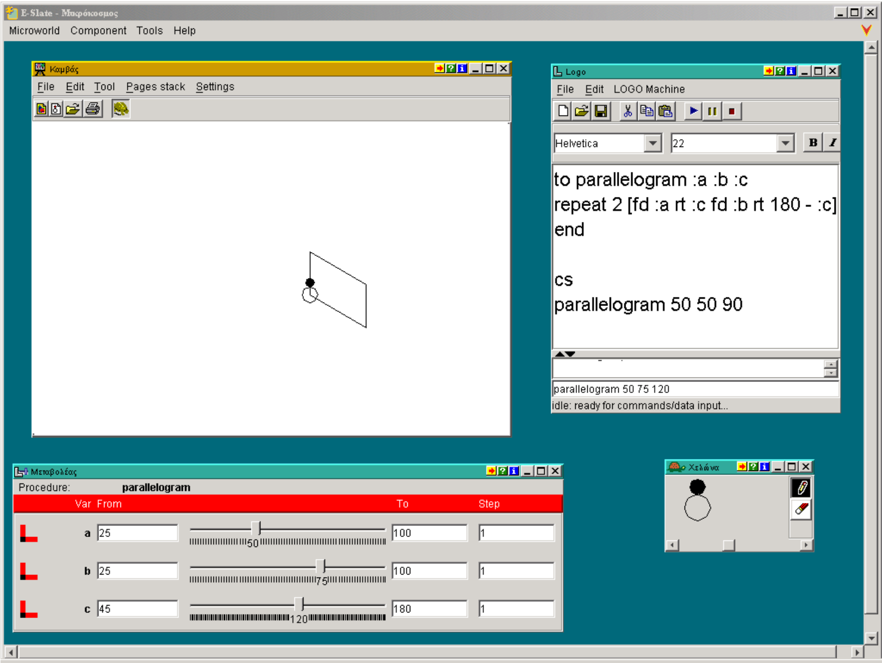

Figure 1: The variation tool

The following sections are vignettes, each of which is based on the design, development and use of a different microworld involving a combination of these components. In each section there are two "modes" of discourse, one focusing on activities and processes and one abstracting from these to raise problematic aspects of the collaboration between the different communities of people involved in the project. References to specific actions and events are made only with respect to the two teams and the software mentioned above. In reality, there were other organizations collaborating in these projects and extrapolations of arguments involving the industry, or the educational reform community have been drawn from that part of our experience.

<u>Vignette 1: Construction of a bar – chart machine.</u>

From early on in the YDEES project, it became apparent that we needed a process for explicitly negotiating priorities in design and development. The development team placed an emphasis on creating the e-slate "desktop", a piece of software which would play the role of the space within which components operate and can be connected. Another emphasis was on developing a small group of components for which it would make sense to connect them in more than one ways. The education team placed emphasis on the mathematics of each components' behavior, on developing some component for mathematical activity and on incorporating the idea of scriptability using Logo as the scripting language. They also placed emphasis on having access to software which could survive in classroom situations and in trying it out in the real school context.

Our jointly negotiated aim was thus to try out the software a) at the earliest possible stage of its development and b) in school conditions which would be as close to normal mainstream school activity as possible. Since the schools were using Logo, a wordprocessor and a graphics package for their computer projects, we decided to develop the corresponding components and the frame within which they could interoperate, i.e. the turtle, canvas, language and the e-slate. Furthermore, we needed to provide our users with something that they could not do with the software they already had. This was the variation tool component.

In our discussions during teacher seminars, our agenda was to suggest a project we had designed on the concept of angle (see following section). Our priority was to have some conceptual control over the innovation so that we could have clear insight into the mathematical meanings created with the use of the variation tool. However, as we were introducing the variation tool functionality by trying out changes to a rectangle with variable sides, the teachers made their own suggestion which was linked to a project they were doing at the time in their Geography lesson. That project was about collecting geographical numerical data and representing it with bar charts. The teachers' suggestion was to use the variation tool to create a series of bar charts for the same kind of data. Each bar would be a rectangle with one set of opposite sides constant and one variable. So, adopting Prawat's (1996) learning community paradigm, we decided to set our angle plan aside hoping that a teacher-initiated activity would help smoother integration of the use of the tool. The project was implemented by five teachers each with their own class. The didactic agenda mediated amongst the teachers and the researchers was for the pupils to construct a “bar-chart machine”, i.e. a piece of software which would create a bar chart, the values in which would be inserted by dragging the slider of the corresponding bar till it reached the required value. In this way, a series of charts could be created, saved and printed in no time. There was inevitably a large variety of ways in which the different groups of pupils went about their project and many did not manage to create the machine (although some did, Kynigos and Argiris, 1999). The most interesting part perhaps in our observations was the many instances of teacher intervention on issues involving rectangle properties, structured procedures and variables.

However, there were important facets to the activity that actually placed obstacles in focusing on the creation of the machine and in the researchers’ analysis of how the component software was used. One facet was that the activity was an extension of a project which did not involve constructions with Logo and was thus affected by not making use of the specific characteristics of the software. In creating a bar chart on paper, for instance, the focus was primarily on getting the axes right, the scaling and the notation on the axes. This has been reported as a characteristic of student perceptions of graphs (Ainley et. al., 2000) Creating the bars would come as the easy conclusion. With this software it was difficult to do so, and in fact it was more difficult even than doing it in Logowriter, the Logo used before, since the functionalities of colour and inserting text on the graphics screen were not in operation yet. There was thus a lot of time and frustration spent on a piece of bar chart creation which was not suitable for making good use of the new software. What we would have liked would be for the pupils to make axes and legends easily and quickly so that they could concentrate on changing bar values and structuring procedures to create the bar chart machine. Meanwhile, the focus for the developers was to get the e-slate functionalities to work and to see the components interoperating. It was completely unimportant, time consuming and distracting to include functionalities such as inserting text and color on the canvas.

So we had a situation where teachers, researchers and developers had different views on the same action line and needed to understand the others in order to negotiate. These views however, were products of three very different priority systems immersed in each job, organization role and history, expertise and objectives. To achieve common understanding so that negotiation was possible would require *hybrid actors*, i.e. people who would have spent the time to gain some insight into the work of the other group. Ideally, all actors should be hybrid actors and in this sense what is needed is a special culture of exploratory software designers, developers and users. However, this takes a lot of time, beyond the scope of a single project, especially when it involves collaboration between organizations and not just the hiring of individuals of complementary expertise. So, the question is, how far can a team invest in integrating expertise, so that the work is delivered on schedule.

Vignette 2: The angle microworld

Our aim as researchers was to design an activity around the concept of angle where pupils might construct and experiment with things which are at changing angles with each other (Kynigos, 1997). In our design we employed the paradigm of the trojan horse by giving pupils a procedure which made the turtle create the classical representation of an angle found in any geometry book. The variation tool, however, turns this representation into a dynamic one where the size of the two "sides" and the angle between them change by dragging the corresponding slider. The aim was for the pupils to start by dragging the sliders in the original procedure and then begin to look inside the procedure, change it and construct a variety of objects (instead of just two lines) at an angle with each other. So we expected to see the construction of models of things like clock hands, a pair of scissors, a pair of walking legs etc. We were also expecting a focus on the angle concept and on the different angular relations between the two objects.

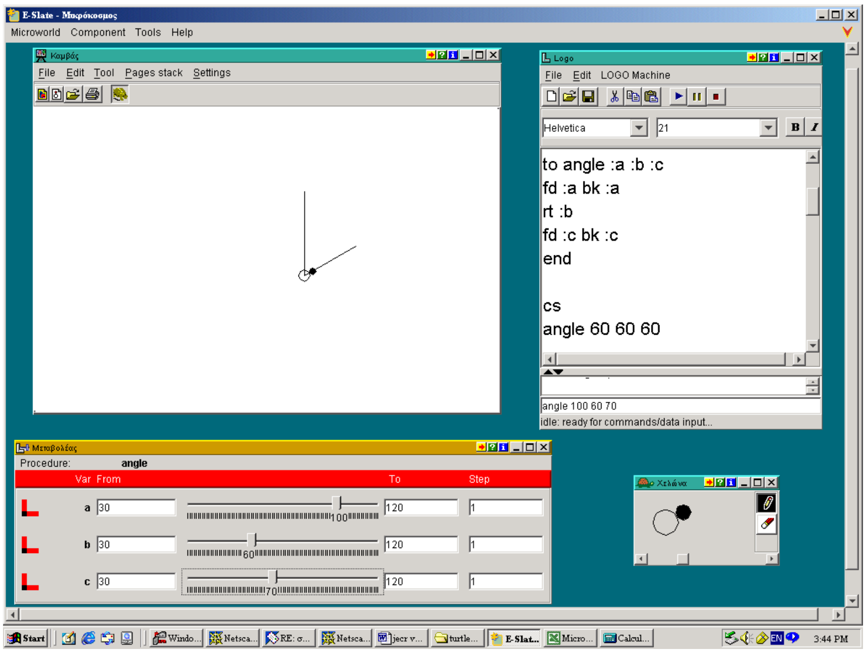

Figure 2: A procedure for an Angle Microworld

This activity was implemented in another school. We held a teacher seminar where we discussed the point of the activity with the teachers and asked them to carry out an angle project of their own in order to get a feel of the meanings which would arise regarding the angle concept. In doing this, we introduced another interface for the variation tool, where for any two variables which could be chosen after executing the procedure once, the change of values could be done in conjunction by dragging the mouse on a 2-D plane, each dimension of which represents one of the two variables. A group of teachers constructed a seagull procedure, making one variable change its size and the other make it flap its wings by changing the angle between them. They then set themselves the task to make the seagull look as if it is flying from inside the screen towards them, i.e. to flap its wings and grow at the same time. Following that, they started to notice the trace of their dragging on the 2D part of the variation tool and discussed what changes there would be in that representation if they were to make the seagull look as if it is flying faster or slower.

They decided it’s the steepness of the lines making the zig-zag shape and then started to ask themselves why.

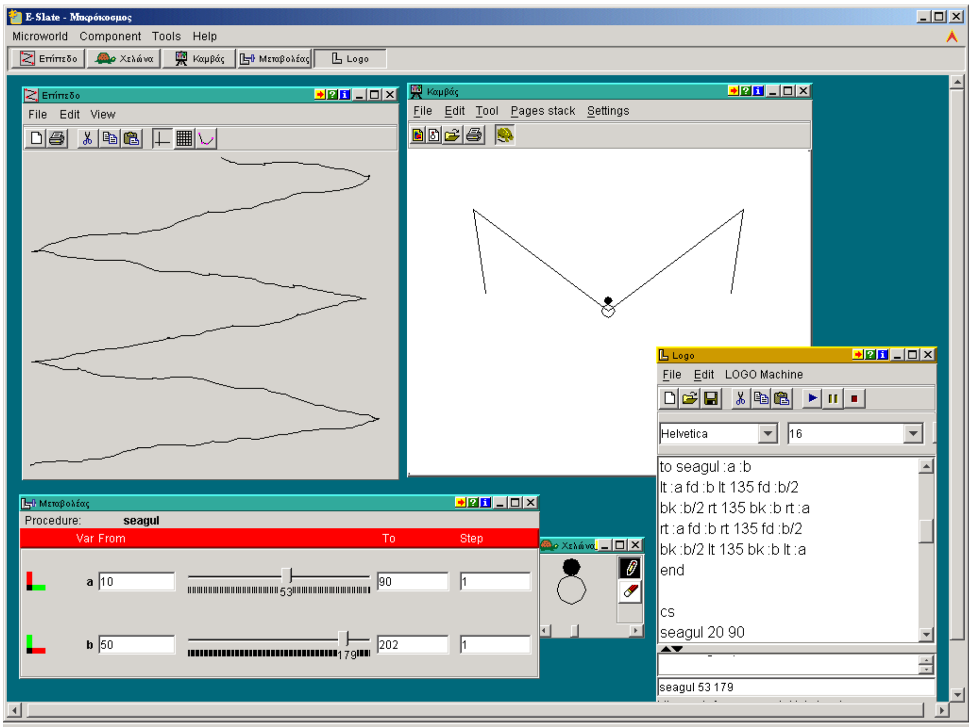

Figure 3: Investigating speed representation with an eagle procedure

At a later stage, during the classroom implementation of the angle project, a large proportion of pupils did not do as we had expected, i.e. work out a precise plan for what they were going to construct by changing the original angle procedure.

Instead, around half the groups in the class made changes to the angle procedure in an exploratory mode without a precise plan to create a specific figure, as e.g. in the x-files procedure in figure 4. In this case, the figure surprisingly transformed into an intricate variety of unexpected shapes (notice the name given to the procedure), providing opportunities for the teachers to intervene with an agenda to encourage reflection on the underlying angle concepts before the changes created an impossibly complicated structure.

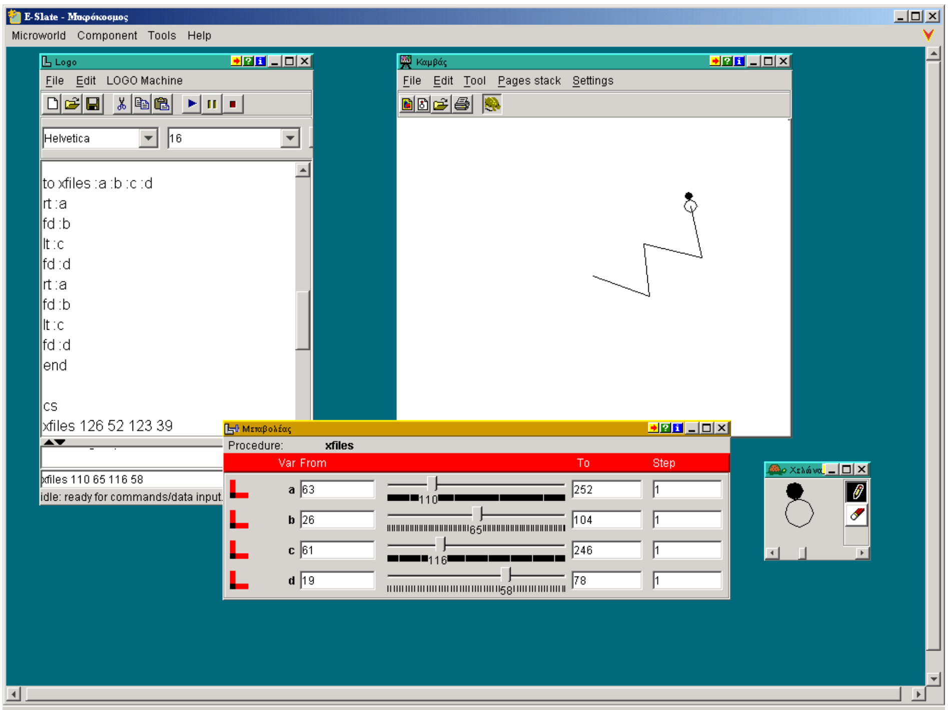

Figure 4. Exploring with the Angle procedure

In procedures such as this, where there were more than one turns, pupils discriminated the dependence of one turn on the other, in the cases where one variable value changed two turns, during the movement of the slider button. This covariation being linear gave rise to discussion on angle operations. In this structure, e.g., the direction of the final segment when moving the variable a (or c) button can be found when adding the first of the affected angles onto the second. The excerpt illustrates an instance where the pupils realized this covariation.

K: Oh, I got it. It’s these two that are changing, these two degrees.

A: So it opens and closes.

K: Yes, like a W opening and closing. Like an accordion.

So these were three snapshots involving the software, the microworld, and the ways it was transformed when picked up by different users. The distinction between tool and instrument made earlier is relevant here. Each of the three cases took the angle microworld and created a different instrument, i.e. a different level and purpose of use, a different construction and a different set of meanings generated through its use. The first was a researcher's design. The second a teacher's investigation and the third a pupil project. It is interesting to contemplate what would be considered as the "product" of this enterprise by different kinds of communities.

The programmer community would focus on the software and particularly on the parts developed from scratch, i.e. the e-slate desktop and the variation tool. They would perceive its use as part of the debugging and testing process. Any dwelling on this use or delays in the production of conclusions concerning what was gained from using it seem like a very large and sometimes unnecessary investment. The research community would consider as product things like the insight into pupil-constructed meanings, teachers' strategies and beliefs or issues related to the social dynamics of the classroom. They would consider the software as an essential tool in this process, and thus any bugs or missing features as an element of research "noise" with respect to the issue in focus. The school community would focus on the pragmatics of carrying out the project, on the topic taught and its association with the curriculum and on the ways it can fit into the daily program. They would perceive the software and the expected learning outcomes as a given. The software industry would focus on how "polished" the product is, i.e. the software and ideally a set of precise instructions on how to use it in the classroom. In fact, it would be more understood if each product constituted a precise description of the content "covered", a different piece of software or tools, and the corresponding material. In the case of Logo with variation tool (not to mention the component world) we have one software for an indefinite number of microworlds, activities and materials and an intention for users to develop their own. The educational reform community would focus on how and where it may fit the existing curriculum, on how many hours the activity needs to last and on what other activity it would substitute. Innovation with respect to content or process of use would be a problematic issue.

Moreover, through any of the above viewpoints, the "product" of work was something which the original designers of the variation tool would have hardly expected. This would be considered as weak planning by most, instead of planning for things to emerge from people who are personally engaged in the "production" process.

This is another issue which emerged as problematic and seen in very different ways: the notion of taking part in an enterprise where things are developing. In R&D institutions this is the way things work. People's roles are to take part in things and/or knowledge which are developing. This means that in any instance in time, it is more likely that things are not ready yet. Thus people cope with uncertainty of how they will end up, planning and vision, paying attention to process and experience. In industry, there needs to be more certainty about the nature and the time of the outcome. Its more like the construction of a building or a car: the task is to make a precise and faultless design and then follow it to the letter. In the school community, there is a perception that teachers are implementers of ready made things like curricula and teaching methods rather than designers or people who try out things (Hoyles, 1992). Involvement in an innovation can thus easily be considered as implementing a new piece of curriculum or method designed by someone else rather than taking part in that design themselves.

Vignette 3: The sinus microworld

The following vignette is about two researchers[6], investigating different ways of perceiving, representing and manipulating constructions based on mathematical concepts. In this case, they were engaged in exploratory activity trying to design a microworld on the *sin* function, based on the notion of constructions where the traces of the turtle and the 2D variation tool constitute two different mathematical representations of the same idea. As in the angle microworld this one is based on a core procedure where pupils carry out some slider manipulation and then look inside the procedure and start testing what happens if they make changes to it.

The design technique in this case is that we take two elements which are related by means of a geometrical property and could thus be constructed by two expressions of the same variable. The procedure is written so that there is a different variable for each of these elements, so that its execution would result in the desired figure only when the two values correspond to the property (i.e. when :y is sin :x in the procedure below). Manipulation of the sliders is carried out to find a set of such values by trial and error and then consider the values or the trace joining them on the 2D slider plane to make some sense of the underlying rule. So, in the tri procedure that follows - which creates three consecutive line segments, two of which have variable lengths and angles between them - variable :x stands for the inclination of the first line segment to the horizontal and variable :y stands for the length of the second segment. For those values that make the figure into a right triangle, :x is the angle at the corresponding vertex and :y is the length of the opposite side. In these cases, :a is equal to sin :x. The first task is to move the mouse on the 2D slider plane to see on which points of the plane we have a triangle. These points turn out to be the trace of the sin function.

```
to tri :x :a
rt 90 - :x
fd 100
rt 90 + :x
fd 100*:a
rt 90
fd 100*cos :x
end
```

In this microworld, there is a connection between the geometrical (trigonometrical) representation of a figure and the cartesian representation of the sin function. The interesting part of it is that while we usually have a construction or a simulation which is represented on a cartesian system, in this case it's the opposite: the user- constructed cartesian representation "creates" the geometrical figure by providing the necessary parameters. Manipulation of the coordinate values on the cartesian plane changes the figure, providing a kinesthetic feeling of what happens when the points are outside the trace of the sin function and how the figure "moves into" a right triangle as the points approach the sin function.

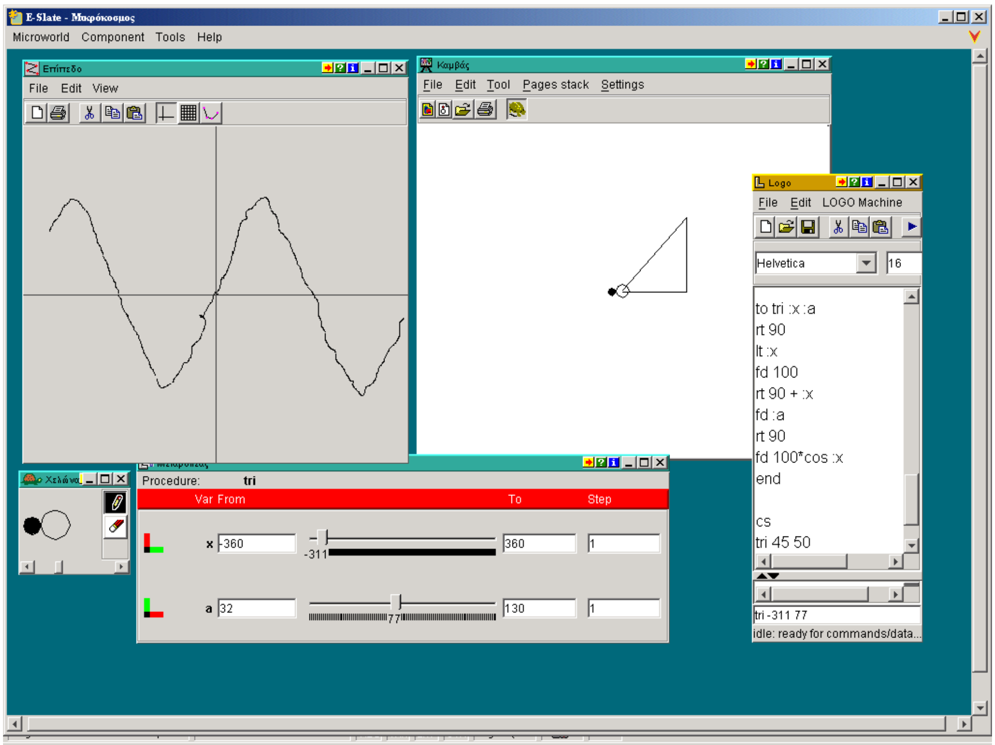

Figure 5: Designing the sin Microworld

There are two issues here which scarcely appear in mathematics curricula. First, the notion that we use an abstract representation to define a figure instead of just to represent it, and second, that we can get a kinesthetic feel of how the figure "transforms" as the points plotted approach the trace of the function which makes it mathematically "neat". We have not yet carried out research on pupils understandings in such a microworld; the point is, how does one "insert" this activity in school, in collaboration with teachers and the school administration since it constitutes a way of doing mathematics which is alien to the curriculum. As researchers, we are “allowed” to question curriculum, method, system and actions. We can suggest new mathematics and new ways of doing mathematics. It is not at all obvious to others that they can be a part of this. In turn, without trying such microworlds in real settings, it is hard to know how they may be put in use within real educational contexts. Furthermore, innovation is not always favorably perceived by industry. Investment in something which requires change in order to be put to use carries an element of uncertainty. In turn, this implies that it is hard for such software to reach high standards from a “product” point of view, reducing its usability in the educational field.

Vignette 4: The ‘Active map & Trip planning’ microworld

In this section the aim is to highlight the problematic of the differing time frames between research, development and the school context and the problem of interdependent actions. The setting was the IMEL project, where the objective was to design collaborative activities over the internet, develop the corresponding components and then implement them and carry out some study on their use. The focus of the University of Athens team was on mathematics and local geography. The activity was for two classrooms to jointly prepare a proposal to exchange visits. The proposal would contain pupil – generated information on the two locales, an analysis of the cost and a schedule for each trip. The issues and the write-up of the proposal would be jointly negotiated over the internet.

There are two important events which happened during the project and which will help bring out the issues in this section. The first was that the researchers, inevitably focusing on the activity, asked for software, the development of which was far beyond the scope of the project[7]. So the activity required a map component where pupils would be able to position and define legends and link each one with information which could then be stored on a database component. That information needed to be represented on a graph component. True to the component architecture philosophy, negotiations were made where components with similar functionality would be used and extended rather than developing everything from scratch as in the case of standalone software. So, database and map components which were already developed would be extended to incorporate the legend functionality and the corresponding interoperability. The graph component would be developed from scratch.

The second event was that a few months into the project, OpenDoc, which was the platform on which the components had been and were being developed, became defunct. So, it was decided that development would in fact begin from scratch on another platform, java.

So we had the following situation: the researchers depended on the software being available in time for some study; the schools had been notified to make teacher time for education and implementation and expectations were built around this by pupils, teachers and administration; the developers had carefully planned so that the software would be ready within a year. In this process, the issue of coordinating and mutually respecting the different timeframes within which each team operated became crucial. There are several aspects of time frames which affect collaboration:

dependency on times fixed by external circumstances (such as the opendoc fiasco or school vacations),

rate with which outcomes are produced,

the ability to make secure and precise plans given the nature of the work and the demand by other partners for secure timing,

what is constituted as progress in time.

It was impossible to shift the whole thing back for six months and carry on from there; teacher availability and enthusiasm was far from guaranteed, school curriculum and vacations had their own rules, researchers couldn't just wait in limbo for six months. Thus there was little choice but to begin the activity by using e-mail and similar standalone software and to substitute this as the project software appeared.

The activity with emphasis on Geography, as mentioned above, was to take place in two phases. The first phase involved the "construction" of a map which included working on a local map by adding information (about the places to be visited) on specific locations. Each time the pupils wanted to add information they chose a location and stuck a legend on the spot in the form of an editable icon. This information was automatically saved on a specific table of a database and was linked to the legend. The second phase was about making the trip schedule and then working out various permutations of travel packages for the cost of the trip, which involved further manipulation of the database data. This consisted of adding new data organized in fields in the database and making charts and graphs of each solution according to cost and number of days. In the project, we decided to begin with the geography activity as soon as the 'map with legends' component and the database component were available rather late in the day. However, we mutually realized that it was impossible under the circumstances to hope for the graphs component. At this point, component architecture and scripting was put in use and actually saved the day.

For the purposes of the mathematical activity, pupils were provided with two tour packages and a range for the available budget. One package had cheaper transport and more expensive hotel rate and the other was the inverse. The point of the activity was for the pupils to insert the information on the database and then plot it on a graph and draw conclusions about which package to go for depending on their choice of number of days and traveling pupils.

Under the project circumstances, we connected the database, language and canvas components and used Logo scripting primitives (cell, field and item repcount) to retrieve package tour data from specific cells or consecutive cells in a field in the database and plot it on the canvas with the help of the standard Logo coordinate primitives as shown in figure 6.

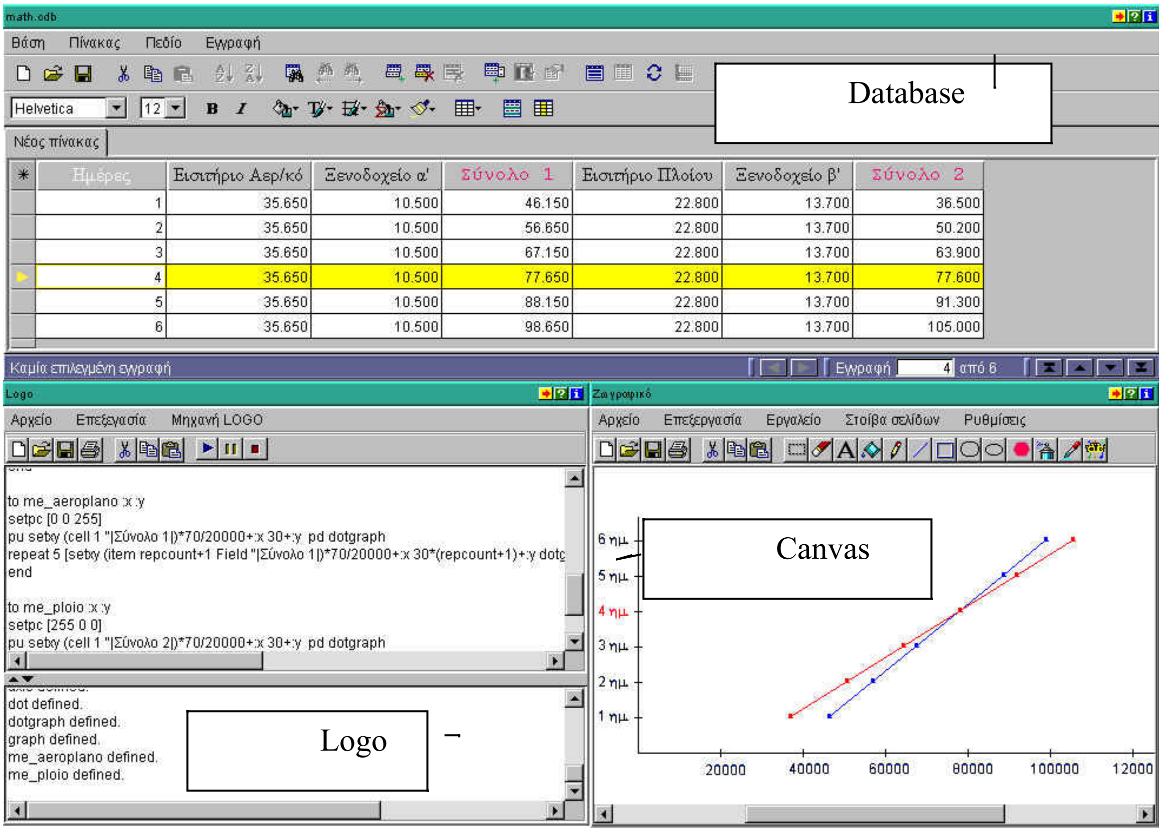

Figure 6: Creating the Database – Graph Microworld

```
to axis :x :y
setpc [0 0 0]
pu setxy :x :y pd
repeat 6 [fd 30 dot]
fd 20 bk 200 rt 90
repeat 6 [fd 70 dot]
me_aeroplano :x :y
me_ploio :x :y
end
```

```
to graph :x :y
ht
axis :x :y
fd 20 bk 440 lt 90
end
```

```
to me_aeroplano :x :y
setpc [0 0 255]
pu setxy (cell 1 "|Σύνολο 1|)*70/20000+:x
   30+:y
pd dotgraph
repeat 5 [setxy (item repcount+1 Field "|
   Σύνολο 1|)*70/20000+:x
   30*(repcount+1)+:y dotgraph]
end
```

```
to me_ploio :x :y
setpc [255 0 0] pu
setxy (cell 1 "|Σύνολο  2|)*70/20000+:x 30+:y
pd dotgraph
repeat 5 [setxy (item repcount+1 Field "|Σύνολο
    2|)*70/20000+:x 30*(repcount+1)+:y dotgraph]
end
```

So, with respect to timing, there was a specific crisis due to a) (external circumstances) and lack of communication regarding b) and c). The only possible solution was to attempt to find courses of action which would allow progress on the school and/or the educational activity front independent of the software development and at the same time permit using new software as it became available. The reusability and scriptability characteristics of the software made this possible since development was just a supplement to what was already available and the canvas component was used to do the job of the graphics component[8].

Conclusions

The issues outlined in this paper are those that emerged as problematic in our experience in taking part in projects involving collaboration amongst organizations and communities of people with different perceptions, expertise and stakes with respect to the integration of exploratory software in education. Say that, ideally the vision in such projects is to have innovative software which has helped integrate an innovation in a niche within the education system. Each community had its own priority system and epistemology (in the sense of diSessa, 1995) regarding the part of the outcome they were involved in and the investment in understanding and respecting the expertise and the role of others. Most of all perhaps, priorities differed with respect to what each community perceived as the outcome in relation to their own trajectory and goal.

Component architecture, for instance, is a contemporary issue in the development of information systems in computer science; had that not been the case, it would perhaps not have been considered worth while by the developers to take this path. Using computers for expression, exploration, construction, information handling and experimentation are within a framework of contemporary pedagogy. The researchers would have been indifferent to (if not offended by) an offer to take part in an Orwelian "I.L.S.[9]" style use of computer science. In this sense, the school based research, the e-slate desktop, the scripting language, the variation tools and the other components are hybrids of these priority systems and epistemologies.

The issue of time-frames is all important. What is considered as progress and as something which can be mediated to others is different within each community. In each case, work is structured in different time spans and is visible at different times. For example, the results of education research may be fully understood two years after the end of a project; the hard part of a piece of software may take three months, to be followed by a year of arduous fine tuning and debugging.

The amount of investment each community makes to acquire understanding of alien expertise is inversely proportional to the time they get on with their own work. Where is the line to be drawn? The discourse used in each case not only involves technical terms but also meanings created only within that community and hardly understood by others. For example, the word “authoring” has a very special meaning in computer science which is hard to grasp if you don’t know the history of the way the user community has been perceived and defined.

Taking part in developing things is also not obvious by all communities. Each would have liked the others to have made up their mind and to keep things constant so that they have some secure ground to build on. Moreover, some communities are used to having a specific number of things unquestioned, stable or changing in a very specific and prescribed way, such as the school curriculum.

In the case of e-slate, the developers’ work has been long term and on-going, since within the persistent goal of continual development of the ‘desktop’ and the construction and re-construction of components, there are all kinds of new technological developments which can be created and used. Within this framework, it made sense to develop the researchers’ community request for the variation tools, i.e. for an idiosyncratic environment integrating symbolic expression and direct manipulation features. The key here is that this environment has survived many hardware, operating systems and technological changes and has evolved in itself through different ideas for links with other components – hooking it onto a database, for instance, was not part of the original functional specifications. On the other hand, the interest of school based research was essential for e-slate use to survive school staff and researcher turnovers, changing labs, different phases in school priorities. A crucial issue therefore, is to *set the right level of generality in the task* to preserve the interest of all the communities involved. What is needed is thus a careful *architecture of objectives*. This is hard to do and is a function of discriminating the rate with which different features of activity and technology are changing. I suggest that this can only be done through the development and preservation of *hybrid communities across organizations*. Uni – organizational or isolated hybrid actor situations are bound to fall short of the demands posed by the complexity of multi – organizational contexts.

Finally, there is the issue of innovation versus instant fit. The former needs harder work, the results are uncertain and the odds for adoption and use in large - scale communities in short time frames are slim. In the latter case, there is the attraction of immediate large - scale use which turns into disappointment; a society will not pick up a new tool to do something which is already being done with traditional technology.

The generation of exploratory software developed within corresponding cultures of developers, practitioners, educationalists and people from industry is not going to automatically come about now that in cases like the European Community, an administrative interest seems to aim to facilitate this. In fact, the history of this community is replete with the inverse scenario of outside communities constructing their own images of exploratory software and its use (Noss and Hoyles, 1996). The constructionist exploratory community has deep understanding and vision of education and the ways it may qualitatively develop, supported with the integration of exploratory software in educational practice. For this to survive and flourish, we need methods of drawing out these and perhaps other problematic issues of collaboration, making them explicit and finding ways to fuel the convergence and synergy between very different communities of people.

References

Ainley, J., Nardi, E. and Pratt, D. (2000) The construction of meaning for trend in active graphing. In *International Journal of Computers for Mathematical Learning, 5,* 85-114. Dordrecht: Kluwer Academic Press.

DiSessa, A. (1995) Epistemology and Systems Design, in *Computers and Exploratory Learning,* A. diSessa, C. Hoyles and R. Noss (eds.), Springer Verlag NATO ASI Series, pp.16-29.

DiSessa, A. (1997) Open toolsets: new ends and new means in learning mathematics and science with computers, Proceedings of the 21st Psychology of Mathematics Education Conference, Finland, Lahti, Pekhonenen (Ed).

DiDessa, A. (2000) *Changing minds*, M.I.T. press.

Edwards, L. (1998). Embodying mathematics and science: Microworlds as representations. *Journal of Mathematical Behavior* 17 (1), 53 – 78.

Eisenberg, M. (1995), Creating Software Applications for Children: Some thoughts about Design, in *Computers and Exploratory Learning,* A. diSessa, C. Hoyles and R. Noss (eds.), Springer Verlag NATO ASI Series, pp.175-196.

Goetz J., Lecompte M. D. (1984) Ethnography and Qualitative Design in Educational Research, Academic Press, London.

Harel, I., Papert, S. (1991). *Constructionism.* US: Ablex Publishing Corporation.

Hoyles C., Sutherland R. (1989) *Logo Mathematics in the Classroom*. Routledge.

Hoyles, C. (1992) Illuminations and Reflections - Teachers, Methodologies and Mathematics*, Proceedings of the 16th Conference for the Psychology of Mathematics Education,* New Hampshire, Vol. 3, pp. 263-283.

Hoyles, C. (1993) Microworlds / Schoolworlds: the Transformation of an Innovation. In W. Dorfler, C. Keitel & K. Ruthven (Eds), *Learning from Computers: Mathematics Education and Technology*, pp. 1-17. Berlin: Springer - Verlag.

Roschelle, J., Koutlis, M., Reppening A., et. al, (1999) Developing Educational Software Components, *IEEE Computer,* Special Issue on Web based learning and collaboration pp 2-10.

Kafai, Y., Resnick, M., Eds. (1996) Constructionism in practice. Designing, thinking and learning in a digital world, Lawrence Earlbaum Associates, New Jersey.

Kynigos C., Psicharis, G. (2001) Meanings on the Notion of Curvature Generated with a Medium for Dynamic Manipulation and Symbolic Expression, *Proceedings of the Ninth Eurologo Conference, Futschek G. (Ed) Lintz, Austria, 155-163.*

Kynigos C., Giannoutsou N. (2001) Seven Year Olds Collaborating to Construct a Map Using G.P.S. and Space Representation Software, *Proceedings of the First European Conference on Computer-Supported Collaborative Learning,* Dillenbourg, P., Eurelings, A., Hakkarainen, K. (Eds), Maastricht, the Netherlands, 364-371.

Kynigos C. (2001) E-Slate Logo as a Basis for Constructing Microworlds with Mathematics Teachers, *Proceedings of the Ninth Eurologo Conference,* Futschek G. (Ed), Lintz, Austria, 65-74.

Kynigos C., Argiris M. (1999) Two Teacher's beliefs and practices with computer based exploratory mathematics in the classroom*, Proceedings of the 23rd Psychology of Mathematics Education Conference,* Israel, Haifa, Vol. 3, pp. 177-185.

Kynigos C., Theodosopoulou V. (in press), Synthesizing Personal, Interactionist and Social Norms Perspectives to Analyze Collaborative Learning in the

Classroom in the Context of a Computer Based Innovation Program, *Journal of Classroom Interaction, 36.2.*

Kynigos, C. (1997) Dynamic Representations of Angle with a Logo - Based variation tool: a case study, *Proceedings of the Sixth European Logo Conference,* Budapest, Hungary, M. Turcsanyi-Szabo (Ed), pp. 104-112.

Kynigos, C., Koutlis, M. & Hadzilakos, Th. (1997) Mathematics with Component-Oriented Exploratory Software, *Journal for Computers and Mathematical Learning,* Editor in Chief, S. Papert, Kluwer Academic Press, Vol. 2, pp. 229-250.

Lagrange, J-P., Artigue, M., Laborde, C., Trouche L. (2001) A meta study on IC technologies in education. Towards a multidimensional framework to tackle their integration. *Proceedings of the 25th conference for the Psychology of Mathematics Education,* van den Heuvel – Panhuizen, M., 1, 111-123.

Noss R (1985) Creating a Mathematical Environment through Programming: A Study of Young Children Learning Logo. PhD. Thesis, published by University of London Institute of Education.

Noss, R. (1992) The Social Shaping of Computing in Mathematics Education. In: Pimm D. & Love E. (Eds), *The Teaching and Learning of School Mathematics.* Hodder & Stoughton.

Noss, R. (1995) Computers as Commodities, in *Computers and Exploratory Learning,* A. diSessa, C. Hoyles and R. Noss (eds.), Springer Verlag NATO ASI Series, pp.363-381.

Noss, R., Hoyles, C.(1996) *Windows on Mathematical Meanings*, Kluwer Academic Publishers, Dordrecht/Boston/London.

Papert, S. (1993) The Children's Machine, Rethinking School in the Age of the Computer, New York: Basic Books.

Prawat R.S. (1996) Learning community, commitment and school reform, *Jnl. of Curriculum Studies,* vol. 28, No. 1, pp. 91-110.

Solloway E. (1998) No one is Making Money in Educational Software*,* *Communications of the ACM,* 41, 2, pp. 11-15.

Projects

YDEES: Development of Popular Computational Tools for General Education: The Computer as Medium for Investigation, Expression and Communication for All in the School, #726, E.P.E.T. II, General Secretariat for Research and Technology, 1995-1998.

I.M.E.L.: Intercultural Microworld courseware for Exploratory Learning, European Commission, Socrates, Open and Distance Learning, 1996-1998.

KALIPSO: Management of the Computer and Communication Technologies in Secondary Education Project, Ministry of Education , E61, Operational Programme for Education and Initial Training EPEAEK, Fourth Support Framework, European Community, 1996-2000.

ODYSSEAS: Integrated Network of schools Educational Regeneration in Achaia, Thrace and the Aegean, Ministry of Education , E11, Operational Programme for Education and Initial Training EPEAEK, Fourth Support Framework, European Community, 1996-1998.

NETLogo: The European Educational Interactive Site. European Community, Educational Multimedia Taskforce, #MM1020, Joint Call in Educational Multimedia, 1998-2000.

Software

Perseus: http://www.perseus.tufts.edu/

Interactive Physics: http://www.interactivephysics.com

Simcalc: http://www.simcalc.com

Tabletop: http://www.terc.edu/TEMPLATE/products/item.cfm?ProductID=39

CSCL – Knowledge Forum: http://www.learn.motion.com/lim/kf/KF0.html

1 http://E-slate.cti.gr

2 In this case, I use the term 'community' to refer to the set of people working with the constructionist exploratory paradigm for mathematical learning (Harel et. al, 1991, diSessa et. al, 1995), rather than to assume an organizational association.

3 Projects YDEES, IMEL, NETLogo.

4 CTI has undertaken the management of a large project called «Odysseia» which consists of around 30 projects involving infrastructure, equipment in schools, software development and localization into Greek, preparation of teacher educators and school based teacher education, supporting actions and five pilot projects incorporating each of the above in a small scale. CTI is the prime partner in the first of these pilot studies called Odysseas. All of these projects are of course developmental and do not incorporate research.

5 Project 'SEED': Seeding cultural change in the School System through the Generation of Communities Engaged in Integrated Educational and Technological Innovation, European Community, 5th Support framework, Information Society Technologies, IST – 2000 – 25214, 2001-2004.

6 Acknowledgement to Dr. Laurie Edwards for a stimulating discussion

7 Not only that, but there were another two groups of researchers in the project, each requesting their own set of components!

8 In fact, in a microworld currently under development, we have Logo procedures feeding numbers to the database and then retrieving them to plot graphs which are then transformed with the help of the variation tool.

9 Interactive Learning Systems, is a sophisticated drill and practice system with a large data bank of questions, fancy statistics on pupil answers and a full network capability so that results can be visible to whoever has the right to see them and most importantly can appear online on the Administrator's screen. They are making good success in the States and the U.K….