ePublishing

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2002)

3ο Συνέδριο ΕΤΠΕ «Οι ΤΠΕ στην Εκπαίδευση»

Generating Communities of Practice for Educational Innovation: Experience from an Institutionally Distributed Integrated Authoring Community

C. Kynigos, E. Trouki , N. Yiannoutsou

Βιβλιογραφική αναφορά:

Kynigos, C., Trouki , E., & Yiannoutsou , N. (2026). Generating Communities of Practice for Educational Innovation: Experience from an Institutionally Distributed Integrated Authoring Community. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 191–202. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/8787

Generating Communities of Practice for Educational Innovation: Experience from an Institutionally Distributed Integrated Authoring Community

C. Kynigos with
E. Trouki and N. Yiannoutsou
University of Athens (NKUA) and Computer Technology Institute (CTI) Athens, Greece
kynigos@cti.gr, etrouki@cti.gr, yiannou@cti.gr

SUMMARY
The paper discusses the experience of generating an institutionally distributed and integrated community of practice within the Greek context and attempts to abstract from this to discuss the main principles and issues in designing methodologies to generate such communities[1]. Our method for generating such a community is to base it on the principle of complementary expertise, constructionism, mutual accountability, idiosyncratic identity and professional development. The community consists of educational technology researchers, software developers, experienced teachers and teacher educators. The function of the community is to co-develop a set of suggestions for innovative use of technology each one consisting of a carefully thought out activity plan and rationale and the corresponding software. The learning domains which seem to be emerging as preference are, Kinematics, Co-variation, Orientation and concept of space, Programming, Control technology, Language and History. Some of the dynamics emerging from our experience as part of this community will be discussed during the presentation, using data from our communications and our constructions.

KEYWORDS*: community of practice, constructionism, authoring, educational innovation*

INTRODUCTION

This paper discusses the generation of a specific type of community of practice, the members of which engage in the design and creation of activity plans and corresponding software tools for educational innovation. The aim is to study the processes of the community's generation and growth and at the same time to consider how this experience might inform educational policy as an alternative strategy for infusing educational innovation with the use of New Technologies. The research is part of a larger project within the European Community Information Society Technologies 'School of Tomorrow' framework, carrying the acronym 'SEED'[1]. The project is currently in its first of three years' duration and it involves three communities, respectively based in Germany, Switzerland and Greece. This paper addresses the experience within the Greek community which consists of educational researchers, teachers experienced in using exploratory software in their classrooms and in teacher education seminars, programmers experienced in developing exploratory software and administrators experienced in managing projects involving the implementation of educational innovations with technology.

The characteristics of the Greek community are the institutional distribution of its members, their complementary expertise with respect to the task at hand, the kind of activity in which they are engaged, which is based on the co-construction of exploratory software and the production of the corresponding documentation and finally the educational, technical and social support which is mediated within the community as professional development for its teacher members. The design principles for this type of community emerged from the lessons learned from our previous experience in attempting to infuse innovative activity through the educational system, by working at the level of school as organization and at the systemic level through the ministry of education (Kynigos and Koutlis, 2002). They are also based on the potential and pitfalls we saw in our own experience of working in such communities within the context of a series of multi-organizational projects (Kynigos, 2002). Through this project we aim to study an alternative strategy of innovation infusion to that of channeling innovation directly through the system and its organizational structure. The strategy involves the creation of more flexible communities of practice working in contexts of less dependence on systemic and organizational pressures and on the subsequent use of the process by which these communities form and grow and the products of the their activities, by making them available widely through the system (e.g. to proliferate as 'exemplary cases and materials').

COMMUNITIES OF PRACTICE FOR EDUCATIONAL INNOVATION

We begin with the principle (argued in Hoppe et al, this volume) that education as a system needs to change primarily because of its ultimate failure to support the generation of meaning and the constructive social learning mindset amongst students. It is not a focus in this paper to analyze the main factors influencing education away from these aims. It is central to the arguments subsequently made, however, to perceive the idea of innovation infusion with the use of New Technologies as a broad task involving deep qualitative changes in mindsets of policy makers, teachers and the society at large of what education can be about. As we point out in the paper mentioned above, with respect to students, New Technologies can at best provide the educational community with sources of information and media for communication and expression. Only if this technology is put to *educational* use, it may support richer learning activity based on symbolic expression, construction, experimentation, investigation, data handling. In such a case, it may also support the generation of social learning modes where authentic questioning, research, use of human and artificial feedback, argumentation become recognized and valued in school. It may thus enable much more focus on learning processes, rather than curriculum content, and on supporting autonomy and creativity in students. With respect to teachers, the educational use of technology may provide them with tools for reflection and systematic professional development throughout their career. It may enable them to try out a much larger repertoire of didactical methods and develop their own pedagogy by becoming professionals with the skills and recognition to look for and to develop their own curriculum and project activities for their students.

Even though these things are possible, however, they involve social and societal changes of the kind occurring over several generations. One of the factors slowing down this kind of change is that systemic and social structures in education are conservative by nature and actually hinder individuals and ad-hoc creative initiatives (Hoyles, 1997, Papert, 1993). A student, for instance, has to conform to the expectations of a system requiring responses to questions on information artificially chosen to portray a fragmented, oversimplified view of the world, often dissociated from reality. A teacher has no time to reflect and design an innovative activity, nor do they have the resources for professional development, which in many ways is not even explicitly recognized. A school administrator needs to portray a picture of conformity to the current social pressure for the school to score high on the traditional exam system. A system administrator has little chance to understand education and is placed in a tight and stringent system with no scope for deviation. A software developer can only perceive of the field of education as one of the target user

communities and anticipate that in his/her career many switches from one target community to the next will be made leaving little ground for real understanding of the potential of educational software to support quality societal development. Software designed to support such activity is generally unavailable due to the high production cost resulting from the required integrated expertise. The organizational fragmentation is an obstacle to this type of expertise. In the case of exploratory software, for instance, teachers are not supported to create their own tools either by the unavailability of suitable software or by the lack of technical support and the knowledge of what is possible with the technology. It is not often that the technology itself is designed to enable or facilitate such use. On the contrary, it is usually designed for rigid layers of access according to technical expertise and a fragmented framework of labor manifested as black – box environments.

Many attempts have been made in the past century for educational reform. There is growing concern, however, that qualitative change in mindsets cannot be supported through the classical reform methods of Total Quality Management, borrowed from corporate reform methods and applied in education (Prawat, 1996). These methods, for instance do not engage the teachers as self-developing participants of the reform, failing to support the generation of teacher communities of practice which in turn may operate as catalysts for new teaching and learning cultures. Grossman et al, for instance (2000) state that teacher communities may support continual intellectual renewal, they may operate as a venue for new learning and for cultivating leadership and that teacher engagement gives out a message to students that genuine engagement with learning is a state of mind and not a theoretical directive. They also describe the difficulties of generating such communities within a single organization (a school) and a uniform type of member (literature teachers). This kind of difficulty has been our experience in several attempts to infuse innovation by collaborating with teachers in their school settings or through the education system (Kynigos and Argiris, 1999, Kynigos 2001). We have found a serious overhead in struggling with institutional constraints such as time for genuine reflection, recognition of implementing innovation. In the current project, we thus decided to focus on the *architecture* of the respective community and on the *type of activity* in which they would engage as methods to 'jolt' its members to dissociate from systemic constraints. With respect to the former, our community is institutionally distributed and consists of members of complementary expertise. The type of activity is the co-construction of exploratory software and corresponding activity plans for students' investigative group project work. In articulating these arguments, we are not making the suggestion that current reform methods should be substituted by this one is implied. However, we suggest it might be interesting to study this strategy as another one in the repertoire of the educational policy maker, so that it enables the potential for genuine innovation, grounded in educational practice, to be made available for inspection and use in the wider educational community.

DESIGN PRINCIPLES

Drawing on previous research on building and sustaining "Communities of Practice" (Grossman, Wineburg and Woolworth, 2000, Wenger, 1998, diSessa, in press), our rationale is that integrated communities may play a role in uprooting people from the constraints posed by the system and their organizations. They may engage them in looking for a means of communication and de-centering of experience. They may enable each individual member to inject enough understanding of alien expertise in order to filter their decisions in design and crafting of innovations. They may thus provide the potential for a synthetic, ecological production of plans for genuine innovation.

The activity of crafting plans for innovative classroom practice involving the secondary development (or authoring) of exploratory software was chosen as a catalyst for this kind of mindset change in the members of the community. Negotiating meanings and terms for

educational change to collaborators of alien expertise working in different organizations requires the re-appraisal, reflection and articulation of a number of issues which in any other case may have remained tacit. Collaborating to construct something tangible such as a piece of software provides the potential for a common ground of reference and for focused, meaningful discussion (for a discussion of the idea of constructionism, see Kafai et. al., 1996). Appropriating technology as expressive media, rather than a mechanism for content delivery or as a tool with a limited set of predetermined functionality enables a much larger community than that of programmers to have deep structure access to the technology (see diSessa, 2000 for a discussion of the term 'expressive media'). We thus perceive participation in our integrated authoring community as engagement in joint constructions to which everyone contributes in a different way – according to his/her own expertise – and still everyone is responsible for others. Interventions made by the research group aimed at providing technical and educational support, motivation, material and communication channels.

Mutual construction

All the participants are engaged in co-constructions (authoring) of exploratory software and written activity plans. Over time, they are expected to create a history of mutual engagement, which will be reflected on shared routines, artifacts and vocabulary. A central facet of participation in the authoring community is the process of creating pieces of software and negotiating on its functionality, its pedagogical principles and the types of activity expected by the students using it. The authoring environment used (e-Slate, http://e-slate.cti.gr) is such that by customizing ready made units of software and by connecting them together, the members of the community can put together creative configurations of functionalities. They do this in a variety of styles and instances, from experimenting in order to get to know the software, to passing little pieces of software around to generate discussion, to constructing more complex software over more time. Constructions are either individual or in small groups. However, the community is aiming to compile them together at the end to make a joint publication/product.

Why authoring?

Authoring involves customisation of software performed by end users without special technical expertise. In Seed, authoring is "the practice" "the shared enterprise"(to use Wenger's terms 1998), the goal towards which the community is working. We perceive authoring to involve not only tools but also practices and pedagogies. In a previous teacher education course, for instance (Kynigos, 2001, E42 project) the process of joining up components, of discussing their functionality and determining the ways in which they might be connected to do something interesting provided mathematics teachers with rich opportunity to engage in reflection and discussion on issues such as epistemology (what is mathematics), learning and teaching methods. On this basis we expect authoring to trigger reflection and change in mindsets as well as to offer the ground for professional development for the teachers.

The context where technology is to be integrated shapes not only its use but also informs and re-shapes its structure (Agalianos 1996). Schools are described as sites with very diverse activities (Rochelle & Kaput 1996). Ready-made educational software cannot support all these activities not only because of their diversity but also due to their dynamic -as opposed to static- nature. A frequent solution would be to add as many features to the software as possible. This option presupposes that all possible uses should be pre-determined, if this was possible it would result in the accumulation of a large amount of features, which makes the software difficult to use because of the amount of information the user needs to learn. Authoring on the other hand empowers users providing them with deeper access to the software which in this case has a few fine-tuned features and generic architecture (to fit in all situations) (Rochelle, DeLaura &Kaput 1996). This combination is possible through authoring because it provides the tools and the mechanism to

dynamically customise the software according to the changing context without requiring special technical expertise. We thus perceive authoring as a common referent for negotiating educational innovation among teachers, researchers, administrators and developers. Authoring is accessible by all because it is designed so that technical and non-technical experts can use it. It is of interest for all members of the community because teachers and researchers can use it to develop software that meets their educational agendas and classroom specificities whereas developers can use it both as a source of feedback (for refining the build-in characteristics of the software) and as means to bring forth special – distinct characteristics of the platform.

Mutual accountability

Mutual accountability is considered to be a central feature of practice as joint enterprise (Grossman, Wineburg and Woolworth, 2000). The project in itself serves as the common framework within which members of the community are mutually accountable. Within the community the outcomes as well as the process belong to all community members. The correlation of ownership and personal engagement is reflected in the following practices. We encourage active participation in self-selected topics: each member decides for themselves the learning area and the concepts on which he/she is going to work. Participants exchange and use others' constructs: they are encouraged to make their ideas or artifacts (activities, tools) explicit to all and help others to use them; the research team cultivates the spirit that constructs are always open to modifications resulting from others' comments and suggestions.

Complementary expertise

The process of developing exploratory software requires a synthesis of expertise: in computer science for software development; in 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. Therefore, we consider complementary expertise as an authentic reason to invest time for understanding others while participating in a joint construction. The different areas of expertise within SEED are:

- Java software development (developers group)
- Secondary development (developers group & researchers group)
- Use of E-slate microworlds in classroom (teachers group & researchers group)
- Scenarios/educational activity design (teachers group & researchers group)
- Pedagogical support (teachers group)
- Academic knowledge of learning theories (researchers group)
- Discipline oriented scientific knowledge

Different expertise serves as a ground for meaningful personal engagement in the sense that everyone feels comfortable to participate in a community of "experts". At the same time working with people of different expertise is a chance to learn from each other and expand the know-how.

METHOD

The rationale behind selecting the members of the community was not immediate and wide replicability, but rather to get together a group of people already experienced in one or more (but not many) of the facets of the process of developing and using exploratory software for innovative activity. Some of the members had previous experience in collaborative settings such as this and some had not. For some, e-mail communication was part of their daily practice, for other it was not. What we requested was that they would be willing to give some time to the project consistently and that they had the technical infrastructure available in their daily work. With respect to developers, we turned to the core team with which we have co-developed E-slate. Also, we invited two developers who, apart from working in a company for educational software, they are working as IT teachers. With respect to teachers, we turned to trainee teacher educators from

the E42 project and to those experienced in implementing project work with such software in their classrooms, in our research site schools. Our community consists of: 10 teachers (7 primary teachers, 3 secondary teachers), 3 teacher educators, 1 administrator, 4 developers and 8 educational researchers. Key points of our method for establishing and sustaining mutual understanding and collaboration are phasing, training and support for software authoring and for designing innovative educational scenarios as well as orchestrating the collaborative work and communication via face-to-face workshops and an on-line forum.

Using an authoring system

In this project the members of the community attempt to author educational software by means: a) of adding content in a ready-made configuration of software components b) of plugging in configurations of components c) of scripting functionalities using "lightweight programming in a simple language" (pp 602 Roschelle et al). The different ways of authoring are presented here with respect to the degrees of freedom offered, so for example adding content in configurations of components is more limited with respect to the customisation possible via scripting. The means of implementing authoring and the degrees of freedom with respect to customisation vary according to the architecture and the characteristics of the platform used. To better illustrate how the community is authoring educational software with E-slate we briefly outline the platform's profile.

E-slate: A black and white box approach: E-slate is an authoring system for developing exploratory software of a wide variety of subjects, functionalities, targeted age groups and levels of use. For the purposes of this project, we are interested in putting to use e-slate's authoring features which allow 'deep structure access' (Di Sessa, 2000), i.e. rather than simply inserting content and defining its form and sequence, so that the community is able to construct structures and functionalities. A core characteristic of the software environments developed with this rationale is their learnability, their 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 the three main characteristics of learnability, all-embracing metaphors and transparency in order to provide teachers and students with ready - made higher - level and technically efficient building blocks, which we call 'components'. The challenge is to see how learnable e-slate authoring is, 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. With respect to those E-slate has adopted a black and white box approach in that it provides technically efficient black box components as higher – order building blocks to build software consisting of component configurations. These components are designed to be as generic as possible so that each can be used in many different configurations and roles (Kynigos 2001). E-slate authoring is not only based on the constructionist paradigm through building component configurations, but also on the connectivity metaphor, providing authors with multiple metaphors for connecting and thinking about component connections. We are investigating how the constructing – connecting combination can support creativity in building software. In this sense, E-slate is based on a 'principled deep structure access' design involving decisions on where to draw the access line in favour of technical efficiency and higher – order functionality constructions.

Connectivity: There are two metaphors for connecting components 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 behaviours. 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. Authoring with E-Slate can be implemented through direct manipulation (plug metaphor), property handling, and/or programming (scripts). This characteristic indicates that the process of familiarisation with the software does not depend simply on accumulation of knowledge (e.g. getting to know all the features and the functionalities) but on progressive immersion into the deeper structure (di Sessa 2000, Eisenberg 1995).

Phasing

We see phasing as flexible scaffolding for the growth in communication and production. Therefore, each phase includes different levels of interaction along with different kind of working modes, configurations and activities. We have a rough plan for these phases, but are open to changes which might seem necessary either to overcome temporary upsets or slumbers in the work, or conversely to allow for the emergence of patterns of communication and work we had not anticipated. The following phases have thus taken place: a) Phase of open exploration: (June - December 2001). Teachers worked individually aiming at becoming familiar with E-slate and authoring. b) Preliminary phase: (December 2001 – April 2002). Community members were clustered in small groups in order to work collaboratively on a specific project of their choice. c) Main phase. During this phase the community will work on the production of software and respective materials aiming to bring these to the level of professional-looking products.

Types of intervention and support

Community's identity is the result of a collective process of negotiation, which does not have to include agreement; illusion of consensus is considered to be the element of a "pseudocommunity", whereas disagreements, tensions and conflicts may promote the establishment of community as long as participants deal with them (Grossman, Wineburg and Woolworth, 2000). Aiming at rich interaction we also provide material and engage participants in a variety of activities for software authoring and for designing innovative educational scenarios.

Interaction within our integrated authoring community is taking place in regular face-to-face workshops and an on-line forum. By now, we have conducted four workshops aiming at engaging members of the authoring community in a variety of activities and participatory styles, in the heart of which is the conjunction of constructing and communicating: panel session, semi structured-discussion, small group discussion, demonstration, hands-on activity. Core points in designing each workshop's agenda are the equal and active participation of sub-communities, the balance between individuals-group, the balance between do-watch-discuss and the balance between microworlds and educational activities.

Training course on software authoring

The main goal of putting together the training course on secondary development was to foster an authoring culture within the community rather than transferring technical expertise to the teachers. Shaping an authoring culture means trying out things, asking for advice and information, presenting and sharing constructions, producing pieces of software that are accessible and –in some cases - useful to the other members of the community, commenting on other people's work and discussing problems, further extensions and other approaches to construction.

The research group decided to conduct the training course on the basis that several members (mainly the teachers and some researchers) were lacking any experience of secondary development and in several cases of using E-slate. A social constructionist approach was adopted

for the design of the course aiming to engage all the members of the community in the process of authoring and to take advantage of the different expertise existing in the community. Specifically, developers have the technical expertise and operate in the black box level (what is inside the components) as well as in the white box (Kynigos and Koutlis 2002) level (what is behind the configurations of components). Researchers have the educational expertise and some of them operate on the white box level. Teachers have experience on educational issues, they know very well the classroom conditions and they are meant to operate in the white box level. (Some of the teachers have experience on using E-slate but none of them had experience with authoring). The challenge was to cope with the different levels of experience and expertise – with regard to the authoring of educational software- and at the same time to unfold the expertise of each group / person in a way that was useful for all the members of the community.

Taking into account the special characteristics of the community and the goal of creating an authoring culture, the research group articulated a set of design principles according to which the training course was organised: a)Providing the resources and the conditions for learning authoring with E-Slate b) Facilitating the development of different learning and working styles c) Fostering interaction among the members of the community and especially among the members of the different groups (teachers, researchers, developers) d) Seeking and facilitating common understanding e) Laying the foundations to construct a common language negotiated by all the members of the community. The implementation of the course was based on a model of direct interaction - communication amongst all the members of the community. In several cases researchers additionally undertook the role of facilitator in the interaction between teachers and developers.

The training course was designed around a "workshop – practice" pattern where face-to-face meetings were followed by electronic communication and personal constructions. Workshops offered a set of basic ideas and principles concerning authoring. Practice periods served a two-fold purpose: a) offered the time to the members of the community to reflect on the authoring ideas / principles and use them to build their own construction b) informed the structuring of the coming workshops. The "workshop – practice" pattern was implemented four times -that is four workshops and four practice periods- in a seven month time (June 2001 – December 2001).

Apart from the emergent issues arising during the practice periods the workshops were organised according to an open ended agenda, which included a set of key issues: 1. Component connectivity: Emphasis on the two metaphors of authoring integrated in E-slate: Plugs and Logo-scripts. 2. Basics of programming with Logo 3.Means of implementing authoring in E-slate: direct manipulation (plug metaphor), programming, component property handling, and event handling 4. Map construction (Map construction is addressed separately because it rests on a very specific rationale deriving from the GIS technology)

From the above issues special emphasis was placed on secondary development with Logo scripts. This emphasis is reflected not only on the workshops but also on the experimental microworlds the community authored during the training course and the subjects of the email communication. Illustrative examples are the agendas of the workshops as well as the fact that a number of experimental microworlds are authored exclusively with Logo scripts even though in several cases the plugs could replace the scripts

Placing authoring with Logo at the core of the training course was a deliberate decision based on two key issues. One was to provide the members of the community with an authoring tool that offered access to the deep – structure and thus supported a larger degree of creativity and component customisation. The other was to engage developers in the process of teaching secondary development and of fostering an authoring culture. With respect to the latter, the developers' contribution on authoring with Logo was considered more crucial and richer in

comparison to their contribution on the plug connections. This hypothesis was grounded on the structure of the plugs, which are specific prefabricated black boxes that cannot be combined in many different ways and thus are limited when compared to the potential offered by a programming language.

Implementation Method:

The training course would rather be considered, as a process of interaction within the community than a series of lectures teachers had to follow. Our agenda for learning the basics of secondary development with E-Slate was incorporated in a series of half – baked microworlds and authoring tools. Those are not ready – made pieces of software to be understood by the teachers, instead they call for modification and customization and thus to gain ownership of the techniques and ideas behind microworld construction. During the workshops members of the research group presented, analyzed and explained the authoring basics and along with members of the developers group helped teachers on hands-on experience of deconstruction, modification, extension and customization of the half – baked microworlds and the authoring tools. In the time between the workshops teachers experimented with authoring and set off to create their own microworlds. At this point the researchers' and developers' group supported the training course via electronic communication. Specifically the developers' interventions focused on technical issues (debugging, explanations on how the components work under the surface, pieces of code that had a specific functionality). The researchers' interventions focused on the pedagogical facet of the training course and attempted to make the microworlds exchanged, tools for learning for the community.

Orchestrating communication

Orchestrating communication could be described as an effort to balance between predefined actions and emergent needs. We present below the key issues in community's communication in correspondence with the design principles we follow.

Mutual construction: We encourage discussions on what kind of information is necessary concerning the presentation of a personal construct (product, process, implications) keeping in mind that each construct may be at a different stage of development. We focus on making explicit that there was no need to share with the community only those microworlds, which we are sure or proud of; it is equally important and useful to share something that was easy to be developed but raises questions or subjects for discussion.

Mutual accountability: We encourage participants to have an impact on community's agenda by introducing alternative topics for discussion, sharing resources, suggesting practices towards facilitating others to understand, learn and construct.

Complementary expertise: We bring to the foreground the discourse each sub-group uses and we encourage everyone to make explicit practices, terms and opinions in order to transmit his expertise. We also stress that helping others with their constructions has to be made by giving hints (providing tools, resources, examples) rather than delivering a requested microworld or just providing educational material.

Also, researchers take advantage of co-participating in order to try to establish norms of communication and collaboration by modeling. In some cases, we try to make these norms explicit by "strategic" interventions, i.e. sending prompting messages and setting up relative discussions during the workshops.

PRELIMINARY RESULTS

For the evaluation of the small-scale community we have collected six types of data: (a) e-mail messages, (b) video (instances) during workshops, (c) field notes, (d) products, (e) interviews and

(f) questionnaire. The researchers work with the community from the inside, adopting the role of participant observers.

In the aim to support the generation of a community, rather than a collaborating group of practitioners, we encouraged all its members to give equal emphasis to both face-to-face and electronic communication channels. In this type of multi-organizational community, we expected a number of communicational and collaboration problems right from the start, such as different discourses, priorities and stakes, modes of communication, acceptance and respect for others of alien expertise (Kynigos, 2002). We were thus careful to bring these into discussion straight away and encourage reflection amongst the group. These issues were manifested by means of the three following issues.

Teachers and teacher educators were "swamped" by the frequency, volume and technical mature of email communication by the developers for whom this was an everyday practice. This was made apparent by a relative quiet from the educators and then a sudden batch of three long emails stating their lack of understanding of the increscent technical jargon, the questioning of whether they belong to this group, their lack of time to read, let alone respond to these. Our intervention was to point out that what we discuss in this community is equally influenced by all and to give some advice for making use of asynchronous communication. For instance, we suggested the following as tips for managing and taking advantage of electronic communication:

- It is not necessary to spend equal amount of time to read or respond to all messages. It is better to select the messages we are going to read more carefully according to our interests and needs.
- It is helpful to others to describe in bullet form our ideas and attach a detailed version of what we want to say.
- It is important not only to respond to existing threads but also to start new discussion threads.
- We become useful to the community when we contribute to the discussion in the area of our expertise. However, the opposite is equally important, i.e. gaining from others' expertise.

The second issue was about the level of detail and the frame of reference by which members presented a microworld they had made during the face-to-face meetings. This issue came up during the first attempt of two teachers to show and explain the microworlds they had constructed. Both they had difficulties in managing to talk only for 15 minutes and slipped to the "history" of their construction without explaining clearly what the microworld was about. The researcher – led discussion that followed resulted in deciding to refer to the following points: a) what is the educational goal of the microworld?, b) How does the microworld function?, c) What are the decisions made on what is black-box for the user?, d) What is the added educational value of the microworld and its use?

A third issue was the breakdown of work. For the members who were inexperienced at the collaborative development and authoring of such software, it was pre-supposed that the roles would be distinct. Educators would give out a 'wish list' of software to be somehow constructed by the developers. We intervened consistently with respect to this to encourage as much as possible the culture that everybody learns to develop software using its authoring features and then use the software as a platform for 'bricolage' over which negotiation and discussion on technical and educational issues would be carried out. As we noticed, for example, that some of the teachers asked from the developers to author the software they wanted, we emphasized the issue "What kind of support should developers provide to teachers"? The reason for this was twofold: a) to establish as preferable practice that developers will give hints and tools rather than "execute" teachers requests about certain constructions and b) to encourage teachers to become active participants focusing on the process and not on the final outcome.

An episode where all the above issues were inherent, occurred between the 2nd and the 3rd workshop when two teachers begun to provide the mailing list with microworlds and one of the

developer responded by sending long messages consisted of scripting. After a long time that this was the main picture of SEEDers e-mail communication (10, September, 2001 – 28, October 2001) three teachers sent messages complaining about their difficulty in understanding the so far exchanged messages concerning authoring. "Are we pursuing one person-band within SEED?" was the question a fourth teacher posed in order to express how worried if she could manage to follow other Seed-ers. What she meant was that if teachers try to learn scripting along with designing educational activities which take full advantage of new technologies and are innovative, then they have no chance be good, because they need to do a little bit of everything. This brought about an interesting discussion on the issue of *hybrid actors* within a community (Kynigos, 2002), i.e. to what extent is it constructive for a member to invest in alien expertise in order to participate and contribute to the production of high quality educational plans and tools. In the spirit of facilitating the collaboration process, however, we did suggest and discuss some specific hints on how to communicate about microworld construction. For example, we suggested the following as possible contribution to threads concerning secondary development: a) the exchange of microworlds (explain what and how, make comments on others' constructs), b) the request for help over authoring, c) the discussion on the aims for constructing a microworld and relative aspects, d) the introduction of general issues or even odd ideas concerning secondary development

We additionally suggested the following aspects for describing a microworld:

- What is the user (teacher, student) supposed to do with it
- What is microworld's added educational value
- Which of the ideas presented in a microworld derived from the "half-baked" microworlds and which (if any) are new
- Which of these ideas seem to be possible for other members of the community to use (extensibility)
- What are the main points of interest (from technical and pedagogical perspective) (verbal description)

CONCLUDING REMARKS

Since at the time this paper was written, we were at the end of our first year in the project and effectively completed around eight months of community work, the experience presented here is not sufficient to make claims as to the success of generating this community of practice. However, each of the issues that came up as problematic seemed to have been discussed and not as yet created visible problems to the community. Furthermore, the productivity of this community has been rather rich, from the point of view of microworlds consisting of around 40 different pieces of software at various levels of sophistication and complexity. An interesting issue which we intent to pursue is the idea of community architectures and types of activity and also with respect to the educational and professional issues discussed and the growing spirit of being engaged in and interesting and pleasant activity. The strategic aim is to facilitate the generation of such flexible communities of practice providing the wider society not just with some exemplary educational content but with some grounded experience of action. Communication technology and software used as expressive media (such as e-slate) may provide us with new ways of supporting this venture. We suggest that careful design of types of participants, an activity designed to bring about the generation of values, practices, expression and products and the initial provision of specialized integrated support may be crucial.

NOTES

1. 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.

REFERENCES

Agalianos A. S. (1997) A Cultural Studies Analysis Of Logo in Education. Unpublished doctoral thesis, Institute of Education, Policy Studies & Mathematical Sciences, London.

DiSessa, A (2002) Boxer progile. Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans

DiSessa, A (2000) Changing minds. Computers Learning and Literacy M.I.T. 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.

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 NATOASI Series pp 175 – 196

Grossman, P., Wineburg, S., Woolworth, S. (2000) What makes teacher community different from a gathering of teachers? Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans, April 2000

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* (1-17) Berlin: Springer-Verlag

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

Kynigos, C. & Koutlis, M.(2002) E-slate a "black-and-white box" approach to component computing. Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans

Kynigos, C. (in press). Generating cultures for mathematical microworld development in a multi-organisational context, *Journal of Educational Computing Research.*

Kynigos (2001) E-Slate Logo as a basis for constructing microworlds with mathematics teachers *Proceedings of the 8th European Logo Conference* Linz, Austria, 75 – 84

Kontogiannopoulou - Polydorides, G.& Kynigos, C. (1993). An educational perspective of the socio-cultural prerequisites for Logo-like education in Greece. In C. Kynigos et al. (Eds.), Proceedings of the 4th European Logo Conference (377-389) Doukas School Publication.

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.

Papert, S. (1993) The Children's Machine, Rethinking School in the Age of the Computer, New York: Basic BooksPapert, S., Harel, I. (1991). *Constructionism.* US: Ablex Publishing Corporation.

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

Roschelle, J., Kaput, J., & DeLaura, R. (1996). Scriptable applications: Implementing open architectures in learning technology. In P. Carlson & F. Makedon (Eds.), Proceedings of Ed-Media 96- World Conference on Educational Multimedia and Hypermedia, American Association of Computers (pp. 599–604) Charlottesville, VA: AACE.

Rochelle, J., & Kaput, J. (1996) Educational software architecture and systemic impact: The promise of component software. *Journal of Educational Computing Research 14 (3)* 217-228

Wenger, E. (1998) Communities of practice. Learning, meaning, and identity. Cambridge University Press.