A HYBRID APPROACH TO COMPUTER SCIENCE EDUCATION

A Case Study: Software Engineering at Aristotle University

Andreas Meiszner, Katerina Moustaka
Institute of Educational Technology, The Open University, MK7 6BJ Walton Hall, U.K.
Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece

Ioannis Stamelos
Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece

Keywords: Open source, Software engineering, Open learning environment, Participatory learning, Open participatory learning ecosystem.

Abstract: Traditionally one characterization of formal education has been that it is 'closed', resulting in the fact that learning spaces with their educational materials, and individual students' learning processes and outcomes remain unavailable for the general public. The hybrid approach to Software Engineering piloted at Aristotle University during the winter semester 2008 / 2009 on the other hand builds upon the way learning and knowledge creation at the participatory web takes place, in particular within the Free / Libre Open Source Software (FLOSS) communities. This is to say that on the hand the learning environment used at this course is open for participation of any individual interested at the subject (inviting in), and on the other hand Aristotle's software engineering students are engaging at students driven small scale learning projects, with each of those learning projects being associated to an open source project (sending out). This combination of 'inviting in' and 'sending out' is what we like to call a hybrid approach. One objective of the hybrid approach is to provide the foundation required for an evolutionary growing learning ecosystem where learning processes and outcomes have the potential to become learning resources for future students and therefore connecting content to discourse.

1 INTRODUCTION

There are a number of challenges for formal education to fully explore the benefits the participatory web provides for education. With regards to collaborative learning and knowledge production the main challenges might relate to the traditional 'closed' and 'semester based' structures of educational systems.

Closed structures on the one hand prevent that students at one institution could engage and collaborate at the web in a 'semi-structured' way with peers from fellow universities or the wider world. This closedness also prevents that the learning resources of the institution might be improved by the outside world, or enhanced through external sources that are brought in by individuals or through technology.

Semester based structures on the other hand provide a challenge to establish a learning ecosystem that would allow for continuous and evolutionary growth; as well on a community level, including the full spectrum of participants ranging from newbies over advanced learners to old foxes, as on a learning resource level. Such a learning ecosystem would on the other hand be desirable as it connects learning resources to learning processes (and related discourse) or the possibility to establish peer support, correction, development or even assessment systems.

A third challenge to education, though not necessarily related to 'closed' or 'semester based' structures, is the question how to provide students with meaningful and motivational learning opportunities that would allow them to develop their professional skills within a real world scenario and impart them as well subject matter skills as also key

Meiszner A., Moustaka K. and Stamelos I. (2009).
A HYBRID APPROACH TO COMPUTER SCIENCE EDUCATION - A Case Study: Software Engineering at Aristotle University.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 39-46
DOI: 10.5220/0001969000390046

and soft skills, such as ICT literacy, critical and analytical thinking skills, project and time management skills, or presentation, negotiation and conflict management skills. To respond to this third challenge web based communities in general, and FLOSS communities for computer science education in particular, might be an adequate equivalent to traditional physical internships, placements or trainings-on-the-job.

So how to address those challenges that prevents education within its traditional structures to fully take advantage of the collaborative learning and knowledge production opportunities the web and the FLOSS paradigm provides?

2 FLOSS COMMUNITIES AS AN EXAMPLE FOR OPEN PARTICIPATORY LEARNING ECOSYSTEMS

To deepen our understanding how collaborative learning and knowledge production takes place at the web we first reviewed at the EU funded FLOSSCom project (FLOSSCom, 2008) one of the likely most mature open participatory learning ecosystems: the FLOSS communities.

Surprisingly the underlying technology used by most FLOSS projects is relatively simple, yet mature, usually including versioning systems, mailing lists, chats, forums, wikis or similar knowledge bases. Additionally free web based services such as Sourceforge provide each FLOSS project with an initial working and community environment therefore facilitating the take off of new projects (Meiszner, 2007).

The way learning takes place in FLOSS is usually a mixture of more than one approach and unlike in formal education learning materials are usually selected by the learner and not the educator. But more importantly, those learning materials are commonly generated by the community itself and also include the code and dialogues between contributors. Further on students are not acting in isolation from previous cohorts of students, but the history of other learners and contributors, and their remaining availability for follow up contacts, constitutes a vital element of the learning materials (Weller & Meiszner, 2008). FLOSS participants also take on tasks such as knowledge brokering (Sowe et al., 2006) therefore taking information and knowledge forward and backward between groups, communities or even language domains.

From a pedagogical perspective learning in FLOSS is characterized by self-studying, project-based learning, problem-based learning, inquiry-based learning, collaborative learning, reflective practice or social learning. It is not assumed that those pedagogies were deliberately set out, but rather that due to the structure, approach and governance of FLOSS communities certain pedagogies have emerged (Glott et al., 2007; Weller & Meiszner, 2008).

Although institutional education might be seen today as the prevalent way of learning, self-education and practical knowledge have their historical foundations long before the institutional formal knowledge. Therefore, communities of common interest like the FLOSS communities, show how exchange and creation of knowledge can be supported by the web in a not institutional way.

As described by Glott et al. (2007) one of the FLOSS characteristic is usually known as ‘openness’ or ‘inclusivity’ of the FLOSS community. FLOSS communities, like any other social formation, have established specific cultural and social patterns and norms that require from anyone who wants to join a certain degree of assimilation. Openness and inclusivity does therefore only mean that those who want to join the community do not have to pass enrolment procedures or have to pass formal performance assessments. Openness also fosters transparent structures as the FLOSS ecosystem is openly accessible, including not only code and documentations, but also communications, discussions and interactions of any kind, e.g. through forums, mailing lists or chats sessions.

A second characteristic relates to ‘volunteering’ and ‘volatility’ since FLOSS participants voluntarily decide which role(s) they want to play or which responsibilities to take on. As a consequence, roles and responsibilities (or capacities) of community members can change over time but also at the very same time depending on the different contexts. This results in a very vivid and volatile internal structure and dynamics of the community (Glott et al., 2007).

A third characteristic is the ‘use of large-scale networks’ and the way they are established and maintained. Besides the individual motivational aspects that must be addressed to attract participants, and to which we will refer later, FLOSS communities enable ‘re-experience’, which is a fundamental mechanism for online learning and knowledge-building (Hemetsberger & Reinhardt, 2006) and also facilitates new member integration. Enabling re-experience and the availability of large-scale networks are also pre-conditions for the FLOSS volunteering support model.

The fourth characteristics relates to ‘content-richness’ and ‘specialisation’. FLOSS communities, though revolving on software development, offer a range of opportunities to participate that by far exceed the scope that is closely related to software (Glott et al., 2007). Content in FLOSS communities provides users with various types of learning resources including manuals, tutorials, or wikis, but also resources that might not be at first recognized as learning resources like e.g. communications, discussions or interactions at mailing lists, forums or chats. One common aspect of the different types of content is that they are jointly generated by users and developers and after generation are overall continuously updated and improved. This however is not limited to a given FLOSS community, but also includes the re-use of artifacts that were produced by other FLOSS communities, or artifacts that are in general freely available through the web. Those external sources are usually brought in to the community by individuals that act as information and knowledge brokers (Sowe et al., 2006).

A fifth characteristic is the aspect of ‘modularity’, which for the FLOSS case reduces systemic interdependencies between different files of the same product, allowing a higher level of task partitioning and a lower level of explicit coordination and interaction among programmers. Modularity might be achieved through a clear division of labour between the core product and more ‘external’ features such as modules, add-ons or plug-ins (Mockus et al., 2000). Within an educational context modularity might be translated to organizational aspects of learning, e.g. to allow participation at a lower entrance barrier, at lower initial skills, or with less time commitment or more efficient usage of time available, or to organizational aspects with regards to modular course design, including resources created by educators and learners.

Learning in FLOSS appears to be comparable with traditional educational settings regarding the underlying technology and pedagogical approaches applied, with one of the main differences residing perhaps on the conceptual and organizational side.

3 POSSIBLE ADOPTION OF FLOSS APPROACHES IN EDUCATIONAL SETTINGS

We suggest three different scenarios on the adoption of FLOSS approaches within educational settings (Weller & Meiszner 2008; Meiszner et al., 2008), with each of them having a different level of complexity and a different degree of benefits:

1. The ‘inside approach’ refers to the practice of taking the principles found in FLOSS communities and applying them within the higher education context. In line with Fischer’s work (2007), this approach involves mapping the key principles onto education, including an evolutionary growth of the course and its environment. This is to say that current students would build upon the work of earlier students developing course and content further year by year, therefore improving content quality and richness and providing regular feedback. Such feedback might refer to course structure, material, processes and tools. The inside approach thus takes the sort of characteristics and tools found in FLOSS as its inspiration. The ‘meta-design’ framework and ‘courses as seeds’ process model is one example for a structured attempt of the inside approach aimed at supporting self-directed learners within virtual learning communities by creating socio-technical environments that support new forms of collaborative design (Fischer, 2007). Fischer (2007) talks of users creating socio-technical environments and has a continuum of participation ranging from passive consumer to meta-designer. This mirrors some of the roles of engagement in FLOSS communities which range from passive users to core developers.

Within the ‘inside approach’ institutions might also decide to ‘open up’ their virtual learning environments to fellow universities or the general public to view what is going on within the environment. Within the inside scenario an institution might even allow those outside groups to participate and engage at this environment, in the case doing so, this likely would be a first step towards a hybrid approach.

A general limitation of the inside approach is that the outside world remains largely or totally disconnected, depending on the degree of openness (e.g. open to view, open to participate, etc.). An example for a semi-open environment is MIT’s Open Course Ware project that is partially open for outside observers, but participation is limited to formally enrolled students only. Another limitation relates to ‘community building’ and ‘evolutionary growth’, since this is *per-se* limited within a given institution that only involves the own student population, and usually even further limited due to (a) a 100% student turnover per semester / course and (b) a comparatively small number of potential community member (formally enrolled students of a course).

The inside approach might be relatively moderate to implement since the technology should be already in place at most higher education institutions, although admittedly modifications very likely would be necessary. On the down side this approach still would keep the students of the institution within this learning environment preventing their semi-structured engagement and collaboration within the wider web. It would also limit the opportunities of 'best of breed', as the wider web might provide better technological solutions or already established and mature communities for respective study fields.

2. The 'outside approach' at which institutions would send out their students into already well established and mature environments to engage at and collaborate within those communities on pre-defined tasks. In contrast to the inside approach, the outside approach might take traditional education as the starting point by providing theoretical information and then sends the students 'outside' to find well established communities, such as the FLOSS ones, to work within those communities and to apply and deepen their theoretical knowledge.

In particular for the area of software engineering this approach might be suitable due to the existence of a large number of mature FLOSS projects and a myriad of educational resources. This is seen in the work of the Aristotle University of Thessaloniki in Greece, where undergraduate students are sent out into real FLOSS communities as part of their degree in software engineering. Students are provided with an initial academic background in principles of software engineering, testing software and the tools and approach in FLOSS communities and then are required to choose and engage with a real project. This clearly has benefits in computer science as it gives students real experience of collaborating with other developers and also of the different types of roles and work required in software development. The outside approach, however, is not restricted to computer programming. It can be realized whenever there is an external, 'real' community that is operating on FLOSS type principles. The case of Washington Bothell University (Groom & Brockhaus, 2007) is a good example for this where students were required to contribute to actual Wikipedia articles as part of their assignment work, thus gaining much of the practical experience of collaboration and authenticity experienced by the software programmers at Thessaloniki.

The outside approach might be the least complex and almost cost neutral; and therefore relatively easy to implement. The benefits of this approach are that it responds to the third challenge as mentioned at the introduction and also would allow for collaborative learning and knowledge production. However, the results of this collaborative learning and knowledge production would remain within this outside community and therefore likely be lost for future students. This scenario would also not provide next year students (newbies) with an easy entrance as no former learners, nor the resources they created, are present at the institutional level to facilitate the newbie entrance.

3. If we view the inside and the outside approaches as opposite ends of a spectrum, then there is clearly a range of blended, hybrid approaches in the middle, which take components of both elements. Such a 'hybrid' approach might be seen as the best option as it allows a continuous evaluation (by educators, students and the wider world) of what 'the best of both worlds' is and how the transferred elements actually suit in their respective new environments. One of the underlying assumptions is that using a hybrid approach, as maybe also partly valid for the inside approach, could be a response to challenges such as a 100% student turnover per semester as (a) not all participating students (and educators) should start at the same time and (b) free learners outside of formal education and practitioners are not bound to any course schedule at all.

Perhaps one such model for this hybrid approach is that of an open participatory learning ecosystem, as outlined Brown & Adler (2008). The concept here is that some of the principles of FLOSS communities are adopted in education (thus it is an inside approach), such as collaboration, use of technologies, or peer production. People learn by doing, for example by remixing or remashing content that is viewed by others. However these activities occur in a broader ecosystem that is open for everyone combining students, informal learners, tutors, experts, organizations, etc, and in this manner it is an outside approach since learners are engaged in a real global community consisting of a range of different spaces. Such a hybrid approach likely would include a number of environments where students could engage at in a semi-structured way and where guidance and support is provided through the use of technologies (e.g. RSS, suggested contents, etc.) and the use of the human factor (e.g. knowledge brokers, community support, etc.).

The hybrid approach also has the potential to open new doors for e.g. (a) new revenue models that could be based in assessment of learners outside of formal education against fees and formal recognition of informally acquired skills, (b) the provision of niche courses and faster identification of potential new courses, (c) up to date learning resources and continuous improvement of processes and products, or (d) an evolutionary growing community including the inherent peer support system.

The drawback of the hybrid approach might be that it probably requires the most drastic overhaul of higher educational practices and might be the most complex to implement.

There are a number of cases within formal education (dePaula, 2001; Groom & Brockhaus, 2007; Wilkoff, 2007; Weller & Meiszner, 2008) that suggest that the 'inside approach' and the 'outside-approach' are viable. Those cases indicate that FLOSS principles can be successfully leveraged to educational settings to provide students with similar learning resources, or allowing them to become content creators. The hybrid model potentially offers the highest benefits but remains to be explored.

4 CHALLENGES FOR THE ADOPTION OF FLOSS APPROACHES IN EDUCATIONAL SETTINGS

There are a number of general challenges such as quality assurance, students' assessment or cultural restraints (Schmidt, 2007) that might prevent the take up of FLOSS approaches within educational settings. As the outside approach has been already applied at Aristotle's Software Engineering course since the academic year 2005/2006, we would like instead to address at this section questions that appeared to us more challenging with regard to the hybrid approach and to which we might not be able to respond within the educational framework we are acting in.

1. The availability of a large number of (volunteering) participants, which is in the case of FLOSS communities characterized by volunteering and volatility, is probably one of the cornerstones of the efficiency of the FLOSS community as a learning environment. A crucial question for transferring FLOSS principles to formal education is how similar networks can be created within formal environments, which usually have small classes. On the other hand, FLOSS community members have regular contacts to only 1 to 5 other community members (Glott et al., 2007) and therefore a question is how to reap similar network effects from small networks in formal education. Meanwhile the 'outside approach' is taking advantage of existing online communities, the 'inside approach' and the 'hybrid approach' will need to establish structures, incentives and motivations to bring together the different involved stakeholders and to establish such a community.

2. How to allow re-experience? Within FLOSS much of the learning processes and outcomes are made visible and therefore allow future learners to learn from what others did and to build upon those experiences – how should this be translated to an educational setting? A project based approach, analogue to development processes in FLOSS, might provide an answer to this as collaboration and discussions could emerge around those project works.

3. The motivational aspect: Motivations to participate at FLOSS are e.g. 'to learn', 'gaining reputation' or 'personal enjoyment', but also a clear 'win / win scenario' between information seeker and information provider resulting in learning benefits for both sides (Demaziere, 2006). Those motivational aspects might be difficult to transfer to and apply in formal educational settings, where the main motivation relates to obtaining a formal degree. While learning in the FLOSS community is efficient because 'project managers' and 'community managers' (and many more roles) voluntarily assume responsibility for organising work, tasks, content, and communication, in formal educational settings roles, tasks, and responsibilities are more pre-determined and rigid (Glott et al., 2007). And even if allowing for such roles within an educational setting, what would be the motivation to assume such roles?

There are a number of possibilities to provide incentives within formal educational settings such as rewards for students who voluntarily assume positions, similar to project or community managers in FLOSS, or to include into the curricula the obligation of more experienced students to share their knowledge with the less experienced. Free learners outside of formal education might also be offered a certification of their learning outcomes against fees, or a virtual credit account that rewards them for taking on roles such as mentor, facilitator, moderator or tutor. Those virtual credits than might be used to pay for assessment and certifications. With regards to incentives for practitioners to participate one possibility would be to involve learner into concrete project works – e.g. to provide computer science students with the opportunity to take on some tasks at a respective open source project. Participants of FLOSS communities are also aware that the skills they learn have a positive value on the labour market and are able to judge this value realistically. Precondition for competing with others that have a comparable formal degree is that informally attained skills in the FLOSS community must be provable (Glott et al., 2007). Peer-reviewing and recognition within the community is very important in this regard to build up a repute that can be shown to possible employers. Similar opportunities, as well for students as for free learners, therefore might be required within an educational setting.

But even if addressing all the points above it might still be a challenge to provide an easy entrance strategy for own and fellow students, or free learners outside of formal education. This challenge relates to questions such as 'what are learners supposed to do' or 'how to get involved'.

5 META-DESIGN & COURSES AS SEEDS AS A SUPPORTIVE DESIGN FRAMEWORK

As a suitable supportive framework for a hybrid approach we identified Meta-design (Fischer, 2007) with its underlying courses as seed process model (dePaula et al., 2001).

Meta-design aims at "defining and creating socio-technical environments as living entities. It extends existing design methodologies focused on the development of a system at design time by allowing users to become co-designers at use time" (Fischer, 2007). Meta-design is aimed to support self-directed learners within virtual learning communities by creating socio-technical environments that support new forms of collaborative design. Meta-design pays tribute to the fact that future uses and problems of socio-technical systems can not be totally anticipated by the design time and must be flexible to changes during use time and allow an evolution through changed or identified user needs. Meta-design pays also attribution to the fact that users might become active participants within a socio-technical environment that bring in their ideas and help shaping and forming the environment and contribute to it. Meta-design is thus describing relatively precisely what can be observed within the FLOSS sphere and was therefore seen to be a suitable supportive framework for the development of a hybrid learning environment.

6 THE CASE OF SOFTWARE ENGINEERING AT ARISTOTLE UNIVERSITY

6.1 Initial Experiences with the Outside Approach

Since the academic year 2005/2006 the 5th semester course 'Introduction in Software Engineering' at Aristotle University of Thessaloniki follows what we have been described as an 'outside approach'.

The duration of the course is 12, 5 weeks and has an average student number of 150 with one of the students' assignments being to participate at a FLOSS project or a proprietary software exercise. In the case selected, the assignment counts for 40% of the total grade. Also, students can work on their assignments beyond the 12,5 weeks of the official lecturing period and submit it at a later time at 3 pre-defined dates per year – by the end of the course in February, or alternatively in June and September.

At the year 2005/2006 15 students volunteered for the FLOSS assignment with the objective of testing FLOSS and to identify bugs.

In the second academic year of the course (2006/2007), which had 24 students opting for a FLOSS assignment, the framework remained the same with the main difference being that students now had two options: to test FLOSS or to develop FLOSS.

At the third academic year (2007/2008) the framework of conducting the course was modified, with the only remaining possible assignment option for students being FLOSS projects, but no proprietary software exercises. Further to this students now had three options: to test FLOSS, to develop FLOSS, or to write a requirement specification documentation for a FLOSS project that still had none. For that academic year 55 students have accomplished their assignment by June 2008.

Motivated by those results, and backed by the theoretical work of the FLOSSCom project, we decided to experiment during the semester 2008/2009 with a hybrid approach.

6.2 Design & Trial of the Hybrid Learning Environment

As part of the FLOSSCom project we developed an experimental hybrid learning environment (www.netgeners.net) and run subsequently a small scale 4 month trial with 10 volunteering students from Greece and Spain, which were located in 5 different countries and supported on a regular base by 1 educator and 2 further less regular participating ones.

This experimental learning environment provided the same type of tools as identified within the FLOSS case and tried to take into account FLOSS particularities such as modularity and project based work. It was aimed to provide learners on the one hand with a basic 'on-board' set of communication and collaboration tools (Blog, Chat, Forum and Wiki) and on the other hand providing a personal space and a space for personal learning projects, including rating and commenting systems as e.g. provided by Amazon.

We than tried to transfer the principle of modularity and project based work through the concept of small students driven learning projects that would allow learners to engage (to a certain degree) within areas of their personal interest; individually or together with other learners as a group work; therefore contributing with their learning projects to the overall development of the learning environment and enhancing its richness up to the point where it might culminates into a very diverse and rich learning ecosystem.

The concept of project based learning projects was also seen as a potential bridge between 'static' content on the one hand and learning processes and activities (discourse) on the other hand that might allow a similar type of 're-experience' as in FLOSS. Learning projects therefore might allow a FLOSS type engagement, where content is often taken forward and backward, contextualized, adapted, translated, re-mixed, embedded into processes or feed into new products by individuals. Those individuals act as knowledge brokers allowing content to be dynamic and causing it to continuously change.

This approach did not intend to provide the learner with a finished set of expert developed 'static' content to be consumed, but instead expects the learner to become an active participant in the respective study field, to acquire subject matter skills through practice, and providing the potential of gaining key and soft skills as a result of their activities and engagement. An underlying believe is that for many 'questions' or 'needs' the answer, or an approximate to it, is already 'somewhere out there at the web' and therefore, instead of 'reinventing the wheel' each time, learners need to learn how to find, analyse, evaluate and use what already exists at the web and to incorporate it into their own work.

Additionally two key aspects of Meta-Design were considered during the design time:

- "A system should be open to change during use time and involves all stakeholders in the design process during design time and use time" (Fischer, 2007).

Though the initial core environment has been largely designed without stakeholders' participation, it allowed for stakeholder modifications from day one of its use time

- "A system should be underdesigned at design time to allow learners ('owner of problems') to create solutions at use time" (Fischer, 2007).

This was taken into consideration by allowing learners to:

- Make use of the communication and collaboration spaces provided 'on-board' or to use any other space at the web that they felt more comfortable with and to link those spaces to the existing learning environment.
- Decide on the objectives, tasks and activities, roadmap of their learning projects and to define its outcomes.
- Provide learners with support and assistance through e.g. regular chats.

6.2.1 Experiences from the Initial Trial

Despite the small group size of participants this initial trial provided a number of valuable information through participants responses to an initial set of questions and two subsequent face 2 face round table discussions.

The obtained feedback suggested that from the technological perspective the learning environment, albeit very simple, responds to the initial needs with the main issues to be addressed being of an organizational nature. Organizational aspects included: more activities that foster community building (e.g. through regular community chats), increased availability of virtual guidance and subject matter support in particular at the beginning, or supportive face 2 face meetings within a class environment.

Some of those aspects should be addressed at our hybrid pilot, which will provide – at least for formally enrolled students – face 2 face meetings and subject matter support. Other aspects, such as community building actions, still need to be taken into account.

6.3 A Hybrid Approach to Computer Science Education – Software Engineering Course 2008 / 2009

Based upon the experiences and work as described above we have been modifying the hybrid learning environment for the 2008/2009 course ‘Introduction to software engineering’. The three options students might choose from remained unchanged to the previous year, namely: to test FLOSS, to develop FLOSS, or to write a requirement specification documentation for a FLOSS project.

Besides the potential benefits of such hybrid learning environment that we outlined above, we do hope that this type of learning environment will provide students with an informal collaboration and cooperation space that is of a practical value to them. This is to say that the initial cohort of students for the year 2008/2009 won’t be able to gain from earlier students’ works, and therefore we must assure to provide by other means, like e.g. regular chats, prompt responses to forum posts, or initial content uploaded by us, that this online environment is of an added student value. Within this, we will also encourage our past year students, which already worked and accomplished their assignments, to participate within this environment and to offer their help to this years students. Such help, as we observed, very often happens on campus and we hope to be able to take part of this discussion online.

Following the hybrid approach our learning environment is open to fellow universities, learners outside of formal education and also open source practitioners, which we hope find some interest in it and join our effort to develop the space further over time in size and scope.

7 DISCUSSION

During this paper we have outlined the rationale behind the hybrid approach to computer science education at Aristotle University, the design approach taken and the initial experiences we gained. We explained which principles of FLOSS and their communities we consider being desirable for educational settings and which might be some of the key challenges to be addressed.

At such an early point it is not possible to predict the applicability of a hybrid approach within the educational structures we are operating at, or what still needs to change. However, having chosen an open design approach, both in terms of methodological framework as well as open in terms of underlying open source solutions, one of the advantages is that we can respond relatively flexible to identified student needs, or the needs of external participants.

REFERENCES

Brown & Adler, J. S. B. a. R. P. A. (2008). "Minds on Fire: Open Education, the Long Tail, and Learning 2.0." EDUCAUSE Review Vol. 43(no. 1): 16–32.

Demaziere, D. (2006). How free software developers work, Môle Armoricain de Recherche sur la Société de l’Information et les Usages d’INternet. 2007.

dePaula, R. F., Gerhard; Ostwald, Jonathan (2001). Courses as Seeds: Expectations and Realities. Proceedings of The European Conference on Computer-Supported Collaborative Learning 2001 (Euro-CSCL 2001), Maastricht, The Netherlands, March 22-24, 2001.

Fischer, G. (2007). Meta-Design: Expanding Boundaries and Redistributing Control in Design. Proceedings of the Interact'2007 Conference, Rio de Janeiro, Brazil, September

FLOSSCom 2008 – Reports, articles and presentations are available at the project website at http//:www.flosscom.net

Glott, R., Meiszner, A. & Sowe, S. K. (2007) „FLOSSCom Phase 1 Report: Analysis of the Informal Learning Environment of FLOSS Communities”, FLOSSCom Project. 2007.

Groom, M. & Brockhaus, A. (2007). Using Wikipedia to Re-envision the Term Paper, University of Washington Bothell.

Hemetsberger, A. and Reinhardt, C. (2006). "Learning and Knowledge-building in Open-source Communities - A Social-experiential Approach."

Meiszner, A.(2007a) “Communication tools in FLOSS communities: A look at FLOSS communities at large – Beyond the development team”, paper and presentation at the Web Based Communities Conference 2007, Salamanca – Spain.

Meiszner, A. et al. (2008) “Free / Libre Open Source Software (FLOSS-like) education transfer report”, FLOSSCom Project. 2008.

Mockus, A., R.T. Fielding, J. Herbsleb. 2000. A Case Study of Open Source Software Development: The Apache Server. Proc. of the Twenty-Second Internat. Conf. on Software Engineering 263 –272.

Schmidt, J. P. (2007) “Open Educational Resources as a higher education strategy for openness and social development” GUNI – Global University Network for Innovation, Newsletter issue September 13, 2007. http://www.guni-rmies.net/news/detail.php?id=1103

Sowe, S. K.; L. Angelis; I. Stamelos. (2006) “Identifying Knowledge Brokers that Yield Software Engineering Knowledge in OSS Projects”, Information and Software Technology, Vol. 48, pp 1025-1033.

Weller, M. & Meiszner A. (2008) “FLOSSCom Phase 2: Report on the effectiveness of a FLOSS-like learning community in formal educational settings”, FLOSSCom Project. 2008.

Wilkoff, B. (2007). Starting From Scratch: Framing Change for all Stakeholders, K12 Online Conference 2007.