



**Queensland University of Technology**  
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

[Dillon, Steven C.](#) (2011) Designing relational pedagogies with jam2jamXO.

This file was downloaded from: <http://eprints.qut.edu.au/47579/>

© Copyright 2011 Steven C. Dillon

**Notice:** *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

## **Designing relational pedagogies with jam2jamXO (Steve Dillon)**

In this paper I want to examine the affordances of the philosophy and practice of open source and the application of it in developing music education software. In particular I will examine the parallels inherent in the ‘openness’ of pragmatist philosophy in education (Dewey 1916, 1989) such as group or collaborative learning, discovery learning (Bruner 1966) and learning through creative activity with computers (Papert 1980, 1994). Primarily I am interested in ‘relational pedagogies’ (Ruthmann and Dillon In Press) which is in a real sense about the ethics of the transaction between student and teacher in an ecology where technology plays a more significant role. In these contexts relational pedagogies refers to how the music teacher manages their relationships with students and evaluates the affordances of open source technology in that process. It is concerned directly with how the relationship between student and teacher is affected by the technological tools, as is the capacity for music making and learning. In particular technologies that have *agency* present the opportunity for a partnership between user and technology that enhances the capacity for expressive music making, productive social interaction and learning. In this instance technologies with agency are defined as ones that enhance the capacity to be expressive and perform tasks with virtuosity and complexity where the technology translates simple commands and gestures into complex outcomes. The technology enacts a partnership with the user that becomes both a cognitive and performative amplifier. Specifically we have used this term to describe interactions with generative technologies that use procedural invention as a creative technique to produce music and visual media. Researching these technologies has been in motion since 2002 applied to iterations of jam2jam software, which was designed for music and more recently visual performance. The software allows users to manipulate a games like interface or external controllers to select musical elements and visual transformations and make choices about the intensity of those concepts through moving icons on an x-y axis up and down and side to side (See for example: (Brown and Dillon 2009; Brown and Dillon In Press; Dillon 2006, 2011; Dillon 2010).

This paper examines the development of an open source version of jam2jam XO for the One Laptop Per Child (OLPC) XO computer as part of a \$468K CRC for Interaction Design (ACID) project called Network Jamming which explored how we can enhance learning and community with creative activities based on collaborative and generative technologies. The project focused particularly on the use of generative systems to increase access to novice users such as children and people with disabilities.

The affordances of digital technologies are present in both the design and experience of technology use, so I will discuss both of these perspectives here and also introduce the voice of the developer in this discussion to present both a philosophical and pragmatic narrative that exposes the potential and the reality of development and application. Music technologies are tightly framed by the musical focus of the system and the nature of extending access to ensemble performance. This raises questions about the teachers’ role and relationships in such a context and the need to evaluate

the educative and social value of the experience with a view to developing strategies for learning. The philosophy of open source is embedded within the architecture of the software and radiates outwards to provide a framework for use. The characteristics of *jam2jam XO* that reflect the OS philosophy are that *jam2jam XO* is:

- **Freely available;**
- Users are encouraged to **explore and share.**
- There is **support for the technology** for learning and
- There is a **mechanism for evaluation and dissemination.**

We could also say that ensemble music experiences with improvisation in music education settings also have these characteristics and we might ask what the technology and indeed open source brings to this transaction? The answer to this lies in an evaluation of the affordances and disruptions that the technology brings to these experiences. The questions for music education are more practical:

- How can we apply the philosophy of ‘open source’ technology in music education?
- What are the relational pedagogies needed for this approach?

In an attempt to bring these issues into discussion I will tell the story of the development process and outcomes of our work with *jam2jam XO* for OLPC and draw out the significant issues and questions into a discussion that raises questions about music education and technology and the relational pedagogies and ethics of the relationship that emerge from our observations and research.



jam2jam XO

### **What is jam2jam XO?**

Jam2jam XO was developed as part of the Network Jamming project for One Laptop Per Child (OLPC) project. This project led by Nicholas Negroponte from MIT Media

Lab was ‘set up to oversee the creation of an affordable educational device for use in the developing world.’ ([http://en.wikipedia.org/wiki/One\\_Laptop\\_per\\_Child](http://en.wikipedia.org/wiki/One_Laptop_per_Child)) Our experiences with software development came from direct contact with Barry Vercoe inventor of Csound musical synthesis language, who heads the development project for OLPC at MIT.

The jam2jam XO software employs the same functions of collaborative performance that we had developed for the jam2jam AV software for Apple Macintosh computers with OLPC computers using a mesh network. What differentiate the projects are the hardware and operating systems involved and the associated licensing process. With jam2jam AV we use Impromptu programming environment that is freely available but not open source. Andrew Sorensen developed Impromptu a computer musician for live coding computer music performances (Brown and Kerr 2009). Consequently development is tightly managed and reviewed with collective feedback from users and the quality of development is assured through cycles of critical development and testing.

Jam2jam XO for OLPC was developed using the Sugar Operating System and designed to be open source, meaning that the source code is free and available to be developed and changed. Thorin Kerr the developer used the Python programming language to develop jam2jam for OLPC XO. With jam2jam XO students can improvise in real-time and play together in an ensemble around musical styles like reggae, country and hip hop, techno. Teachers primarily use it as a hub for instrument performance or singing/rapping and often as a basis for literacy numeracy activity. From an open education perspective the system promotes collaborative learning in an accessible way and frames successful outcomes for the students because of the focused musical styles.

### **The Education goals of Sugar**

jam2jam XO is based on an open source operating system called Sugar whose approach hinges on the idea that it ‘is useful only to the extent it is used by the learning community. Sugar Labs works with educators around the world to focus on these learning challenges:

- To make Sugar and Sugar activities **freely and readily available** to learners everywhere
- To **explore and share** best practices
- To provide a forum for discussion and **support for technology for learning**
- To provide a mechanism for **evaluation and dissemination** of results

(<http://sugarlabs.org/index.php?template=page&page=contributors>)

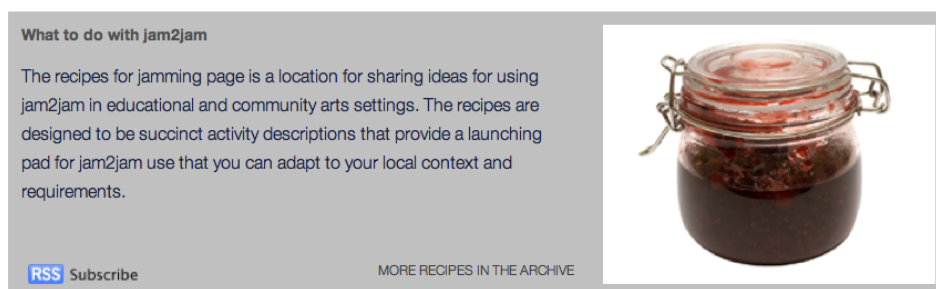
The idea of open source in technology development alerts us to the affordances of these practices outlined by Sugar. Most clearly it is about drawing upon cognitive surplus (Shirky, 2010) to collaboratively develop the source code. We can see here

congruence with Deweyan notions that were popular in the 1970's (Dewey 1970, 1989). jam2jamXO is open source in the several ways; firstly the source code can be changed. Secondly new musical style scenes can be composed and shared. Further opportunities for sharing and developing learning experiences come from the jam2jam experience design ethos that utilizes the metaphor of learning *recipes* See: <http://explodingart.com/jam2jam/jam2jam/Recipes/Recipes.html/>). Teachers and community musicians share ingredients, processes examples and nutritional value (knowledge or social outcomes).

### Sharing Recipes

Out of these three means of participating it is only the last that has occurred so far. Changing source code is complex and specialized. Changing MIDI files is accessible for the techno-savvy teacher but in our experience so far this has mainly been consumption rather than participation in the process. What has happened is teachers and music educators, artists and curators share recipes or rather provide video clips of the product and supply a brief outline of the experience design and any associated resources.

### recipes for jamming



What to do with jam2jam

The recipes for jamming page is a location for sharing ideas for using jam2jam in educational and community arts settings. The recipes are designed to be succinct activity descriptions that provide a launching pad for jam2jam use that you can adapt to your local context and requirements.

RSS Subscribe

MORE RECIPES IN THE ARCHIVE

### Technical challenge

As researchers we have been faced with the challenging task of developing a collaborative ensemble experience using a processor that has slightly less power than a smart phone. However development was made and the software created and made freely available. Downloads in June 2011 exceed 30,000 and this figure may be amplified by teachers who copy the software to class sets of computers. To date no activity in scene development has been made. Recipe projects have abounded with sharing of Recipes happening in Sweden, Norway, Australia and Uruguay. Most of the activity in making recipes has been generated by undergraduate education students in developed countries and teachers in developing countries.

What this says about technology and Open Source is that just because the code is available to be changed it does not mean it will be. In relation to technology even the development of a few simple MIDI files is also available the technological expertise and perhaps the fear of technology is still deeply embedded in the teaching experience. Perhaps due to the perception of complexity. However what really

motivates educators is a good idea for learning that can be exchanged quickly and easily.

### **The developer's voice**

Thorin Kerr had similar remarks to make from the developers perspective:

Open source development is really a mixed bag for a developer. Developing open source software can take on an ethical and political standpoint. The software is being developed for the good of the community. This could be a community of software users, but it also includes a community of developers. Open Source software offers it's intellectual property for others to learn from and build on. This means, observing the particulars of the license, developers can contribute or modify the software, or even use parts of the software for their own project. In short, by volunteering their time and skills, the developer can feel that they're making the world a better place. They can achieve recognition amongst their peers. The code of their software can be studied and analyzed, meaning it can justifiably be said that the software is a contribution to knowledge. There is also the notion that open source software is collaborative. Once open source software is made available, anyone else is to take up the baton of further adapt and develop that software. These are all noble and compelling reasons for developers to invest their time and skills in an activity which usually doesn't directly lead to any financial gain.

However, the practicalities of developing in the open source world can be frustrating, for a few reasons:

Every developer relies on accurate and comprehensive documentation of the tools they use to develop with. Documentation for many open source tools is poor. There are various reasons for this. The documentation may be an open source project in itself. Wiki's can spring up with error prone or conflicting information. Sometimes the documentation can seem to be written by experts more interested in demonstrating their knowledge than assisting a newcomer. Documentation can also be generated automatically producing obtuse results. More generally, while most software developers acknowledge the importance of documentation, producing quality documentation may not be a skill they possess, or have the inclination to spend time doing.

All software faces the prospect of obsolescence. Software relies on numerous resources which are constantly evolving. Hardware changes,

operating systems change and as a result software must adapt to continue to work. However at times it can seem the resources available to the open source developer are conspiring to ensure their software will fail. It is common for Open Source software to lean on numerous other open source resources. Typically these are called 'dependencies'. For example, Jam2JamXO relies on the open source tools available in the Sugar environment, Python, GTK, OLPCGames and Csound. In turn each of these tools rely on a multitude of open source tools (Pygame, SDL, libsndfile and ALSA to name just a few). With open source software all these dependencies develop along their own path. The reasons for these changes in the open source world can be arbitrary. There is no co-ordinating organization to ensure compatibility or consistency with any other component the software relies on. The tendency for things to break over time is high. Sometimes, software libraries may be left abandoned and unsupported - not because they weren't widely used, or important components - but simply because only one individual was developing that library, and they simply decided to do something else. Of course, with these tools being open source, there is the freedom to take up the maintenance of these components for oneself. However, this can be daunting, and largely distracting.

The ideal that others will pick up development of your software, or that a community will grow around your software is something of a rarity. There are large open source tools with active user communities. However, even these projects may only have a handful of developers. Invariably, it may be that your software does not have the impact or noble contribution that you thought it might.

However, perhaps this is a good thing. Perhaps the reasons to develop open source software shouldn't be based on recognition from peers or changing the world. Instead, developing open source software should be for the pure joy of developing software. I suspect this is what drives most open source projects at the moment. It's this non-product oriented direction for open source software which actually gives it a unique social and cultural - and non commercial - worth.

Thorin provides further critical considerations here about open source that need to be taken into account. Firstly the idea that just because you can change the code doesn't mean that the changes will be of high quality. Secondly, the documentation about the software may not be accurate so that development can be impeded by lack of critical information about how to change the software. Finally the reason for development benefitting community may not involve widespread uptake and such uptake is not common. However Thorin's last point from the point of view as a computer musician is that the act of creating code is itself intrinsically motivated and the benefit is personal. These negative aspects aside the development of

jam2jam XO has been successful in its uptake as demonstrated by downloads. From a technical point of view the idea of improvisation on a mesh network with up to 4 computers each taking an instrument role provides a kind of ‘*Switched on Orff*’ metaphor for using mobile technologies for learning music and learning through music. The metaphor draws upon the Orff Schulwerke approach to learning music through using junior versions of instruments (Computer models of bass, drums, chords and lead) and cultural materials (Musical styles as midi file ostinato). I will remind the reader here that most music technology is used in the production rather than the performance of music. This model is innovative in that it is about improvised performance with computers in real time. The open source transaction here is represented in the philosophy embedded in the Sugar operating system. The very act of improvisation is about exploring and sharing music through performance.

There is some **support for technology for learning** with the community sharing its experiences with the developers and research leaders and amongst themselves. The mechanism for **evaluation and dissemination** of results has also been built into the projects sustainable outcomes. Even after a year since the funding ceased there is activity within the community that exchanges recipes and tips, reports faults and bugs and simply reporting happy uses of the software across the world. As Thorin suggest the pleasure for the developer may come simply from the intrinsic challenge of making the software and sharing it, likewise from a music educators view the pleasure comes when I see creative and often unexpected applications of the software as well as simply smiles on the faces of children as they enjoy the experience. There is no doubt to me that the philosophy of open source is embedded within the architecture of the software and radiates out to provide a framework for use. jam2jam XO is freely available. Users are encouraged to explore and share. There is support for the technology for learning and there is a mechanism for evaluation and dissemination.

### **Conclusion**

Open Source like pragmatist philosophy hinges upon experience and intrinsic engagement. In software design we can embed philosophical principles such as those that are present in Sugar. An observation of this has been that technology can frame and focus experience in clear ways and encourage particular behaviour. With jam2jam the focus is upon performance and collaboration:

*‘This gives me the perfect tool for working with cooperation and socialisation’* (Per Skold Humfryskolan Malmo Sweden:

[http://explodingart.com/jam2jam/jam2jam/Home/Entries/2010/8/20\\_Communal\\_Creativity.html](http://explodingart.com/jam2jam/jam2jam/Home/Entries/2010/8/20_Communal_Creativity.html) )

The technology extends the capacity for performance and expression and facilitates complex interactions in accessible ways. Open source is the free form improvisation of the computer-programming world. Is it any wonder then that there is a cross over in the philosophy and practice? Is it also any wonder that this is a world occupied by a few brave individuals that constitute a community of risk takers and creatives?

So what does this mean for the two questions raised earlier:

*How can we apply the philosophy of ‘open source’ technology in music education?*



What these experiences suggest is that it is these qualities of exploration, sharing and documenting that should be on our agendas in the development of learning experiences both about music and through it. Whilst these are fundamental to pragmatist philosophy through discovery and experiential learning we have to ask what dimension does this technology with agency provide? The answer here is simply *access* consequently the embedded knowledge within agent technologies that are available allows us to think about learning experiences in more complex ways and also more importantly more concrete ways. A young child can experience making complex music concrete timbral works in real time with such technologies where as in the past this would be a passive listening experience. With jam2jam XO young players can perform synchronised electronic music in real time at age 5. So for teachers we have to consider how we can transform this concrete and complex activity into knowledge and understanding. How do we move it from embodied understanding to a reflective discourse? There is a need to reconsider the evaluation of experience also and include demonstrations of knowing and reflection not just in and on action but through it in a conversation with an artifact of the musical experience or demonstration present.

*What are the relational pedagogies needed for this approach?*

The power differential with these kinds of technologies and the open source philosophy are also reminders of experiential learning frameworks. They centre around providing opportunity for meaningful engagement to: explore, attend, evaluate, direct and embody in personal social and cultural contexts ([http://explodingart.com/jam2jam/jam2jam/Research/Entries/2008/1/12\\_Meaningful\\_engagement.html](http://explodingart.com/jam2jam/jam2jam/Research/Entries/2008/1/12_Meaningful_engagement.html)). This requires a teaching and learning relationship that is inclusive and based around collaborative activity that frames and focuses the learning. The teachers role in this is to draw out the learning experiences engage in a musical discourse that seeks to revisit fundamental concepts at progressively deeper levels (Bruner 1966; Bruner 1986). The difference here between the kind of learning advocated by Dewey and Bruner is that the technology both frames the learning and also provides a potentially intrinsic activity and access. The teacher's role here involves a clear understanding of the affordances of the technology and the knowledge that is framed by it and the experience. The teacher a partner in the learning process and the technology fulfills part of the role of engaging, focusing and binding experience. The teacher is concerned with how to synthesize and make sense of experience.

The philosophy of open source is already deeply rooted in democracy in education. What open source technology adds is the potential to apply these values in design of software for learning. Sugar OS questions the idea that we use operating systems designed for business for education. Their response is to make an OS that is fundamentally collaborative. Regardless of uptake and development by community

the teacher's role is to interpret the affordances and seek the opportunity to widen access and deepen experience through the application. Open source as the free form improvisers of the development world will be the source of many discoveries and provide perhaps an idealistic and utopian vision. Even after sixty years of experiential student centred learning, mimetic approaches are still the predominate mode of delivery of education. Drill and practice and surveillance software bases their design on these models of interaction with students. Open source provides us with a design model that includes exploration, collaboration and clear evaluation. We need to be conscious of these in our appraisal, selection and application of software for our classrooms.

## References

- Brown, Andrew, and Thorin Kerr. 2009. Adaptive Music Techniques. In *Improvise: The Australasian Computer Music Association*. Queensland University of Technology, Brisbane, Australia. : ACMA.
- Brown, Andrew R., and Steve Dillon. 2009. Jamskolan09: An International workshop and Symposium on arts Education through Network Jamming with Generative Media. Brisbane, Australia: Australasian Cooperative Research Centre for Interaction Design (ACID).
- Brown, Andrew R., and Steve C. Dillon. In Press. Collaborative Digital Media Performance with Generative Music Systems In *Oxford Handbook of Music Education*, edited by G. Macpherson and G. Welch. New York: Oxford University Press.
- Bruner, Jerome, S. 1966. *Towards a Theory of Instruction*. Cambridge Mass: Belknap Press of Harvard University.
- Bruner, Jerome S. 1986. *Actual Minds , Possible Worlds*. Cambridge Mass: Harvard University Press.
- Dewey, John. 1916. *Democracy and Education*. New York: Macmillan.
- . 1970. *Experience and Education*. 12th Edition ed, *The Kappa Delta Pi Lecture Series*. USA.: Collier-Macmillan. Original edition, 1st Edition 1938.
- . 1989. *Art as Experience*. (1st Ed.1934), 1980. ed. U.S.A.: Perigree Books.
- Dillon, Steve. 2006. jam2jam: Networked Improvisational Musical Environments. In *School Music and Teacher Education: A Global Perspective in the New Century*, edited by M. Moore and B. W. Leung. Hong Kong: Department of CAPE, the Hong Kong Institute of Education & International Society for Music Education (ISME)
- . 2011. Access to meaning: Symbolic Interaction and relationships. In *Learning music through online media performance: Network Jamming with jam2jam*, edited by A. R. Brown and S. Dillon. New York: Routlege.
- Dillon, Steve C. 2010. Access to meaning: symbolic interaction & relationships (Virtual Buber). In *International Society for Music Education: . Beijing, China*.
- Papert, Seymour. 1980. *Mindstorms, Children, Computers and Powerful Ideas*. USA: Basic Books Inc.
- . 1994. *The Childrens Machine: rethinking School in the Age of the Computer*. New York: Harvester Wheatsheaf.

Ruthmann, S. Alex, and Steve C. Dillon. In Press. Technology in the Lives and Schools of Adolescents In *Oxford Handbook of Music Education*, edited by G. Macpherson and G. Welch. New York: Oxford University Press.