# A Behind-the-Scenes Peek at World's First Linux-Based Laptop Orchestra – The Design of L2Ork Infrastructure and Lessons Learned

**Ivica Ico Bukvic**
Virginia Tech, DISIS, L2Ork, ICAT
Department of Music - 0240
Blacksburg, VA, 24061
ico@vt.edu

## Abstract

In recent years we've experienced a proliferation of laptop orchestras and ensembles. Linux Laptop Orchestra or L2Ork founded in the spring 2009 introduces exclusive reliance on open-source software to this novel genre. Its hardware design also provides minimal cost overhead. Most notably, L2Ork provides a homogenous software and hardware environment with focus on usability and transparency. In the following paper we present an overview of L2Ork's infrastructure and lessons learned through its design and implementation.

## Keywords

Linux Laptop Orchestra, L2Ork, Infrastructure, Lessons Learned

## 1 Introduction

Laptop orchestras arguably need no introduction. Their number is now growing at an exponential rate with new ensembles being introduced at Universities and other independent institutions across the world (“International Association of Laptop Orchestras,” n.d.). With the ensuing growth, the laptop ensemble repertoire is increasingly hampered by the lack of software and hardware standardization. The mounting cost of obtaining and/or building supporting infrastructure has resulted in compromises, further exacerbating the said fragmentation (I. Bukvic, Martin, Standley, & Matthews, 2010; Kapur et al., 2010; Trueman, Cook, Smallwood, & Wang, 2006; Wang, Bryan, Oh, & Hamilton, 2009). One notable aspect of ensembles based on the PLOrk model (Trueman et al., 2006), is the use of hemispherical speakers whose purpose is to provide co-located sound. Prior to the introduction of the hemispherical speakers, the computer music genre has primarily relied on either acousmatic spatialization or a more traditional stereoscopic means of sound reproduction. Both of the said approaches pose a challenge in that they can disassociate the music-making process with its source. In such a setting typically one or more computer musicians located on stage making (or mixing) sound are being heard across a vast array of speakers, none of which correspond with their physical location. This is where hemispherical speakers make a considerable difference in bridging the said cognitive gap (Smallwood, Cook, & Trueman, 2009) and reintroducing the critical notion of co-located sound inherent to acoustic music instruments.

## 2 Introducing L2Ork

Linux Laptop Orchestra or L2Ork (I. Bukvic et al., 2010) was founded in Spring 2009 as part of a research initiative with focus on providing a discipline-agnostic point of convergence for arts, technology, and engineering. The initiative has gained considerable traction, attracting twenty on-campus stakeholders and four corporate sponsors. In part due to its open design, and in part due to its ambitious goal of complementing K-12 (elementary school to high school) education by introducing this unusual integration of arts and sciences, L2Ork has also received an unprecedented amount of media attention, including the front cover feature in the Linux Journal (Phillips, 2010). By partnering with the regional Boys & Girls Club (an after-school program for inner city children) and through the help of a series of grants the project has produced a smaller complementing satellite laptop orchestra for the purpose of training 4$^{th}$ and 5$^{th}$ graders in the newfound art of making laptop ensemble music

(Ivica Bukvic, Martin, & Matthews, 2011). Since its inception, the ensemble has had four major tours, including a twenty-one-day tour of Europe with stops at STEIM and IRCAM. Last February, a work titled *Half-Life* won the first place on the first international laptop orchestra composition competition (Elliott, 2011). Apart from the fact that the ensemble is a result of academic research and a part of academic curriculum, one of its core goals is also to establish L2Ork as a professional and active ensemble outside the academic circles.

By its very design, the ensemble attracts students from various backgrounds, amounts of musical training, and/or computer literacy. For this reason, a part of the project's focus is on the development of optimal interfaces for the delivery of critical musical cues with the assumption that users have no prior musical experience. This challenge is further amplified by ensemble's focus on physical presence and choreography, most recently with the inclusion of Taiji (Tai Chi) mind-body practice and system's ability to "harvest" ensuing motion data and translate it into meaningful musical cues audience can clearly associate with.

Four unique, yet mutually dependent components that in many ways define the ensemble's approach to the laptop orchestra genre are:

1. interest in developing physical presence and choreography;
2. exclusive reliance on open-source;
3. affordable design, and
4. focus on homogenous environment and optimal usability.

In the following section we will visit some of the most notable challenges and solutions to problems related to both the genre and the Linux platform.

## 3 Homogenous Environment

One of the challenges in maintaining a laptop orchestra is the uncertainty caused by the use of individualized software platforms. Even for ensembles relying on fairly homogenous environments, such as Windows and OSX, hurdles remain in terms of installed software, some of which can at times collide with each other and cause compatibility problems. Such challenges can easily waste a majority of time allocated for a rehearsal on seemingly trivial things, often resulting in a frustrating experience. There is clearly a need for a turnkey solution, a system that simply does what it is meant to do. The author argues that chances of achieving such an environment, particularly within the Linux ecosystem, is to ensure that the OS and supporting hardware are truly homogenous. L2Ork achieves this by essentially loaning out fully preinstalled and preconfigured systems together with supporting hardware to students/users who are expected to treat the ensuing amalgamation of hardware and software as an integrated instrument. This is where the affordable design plays a critical role. With each seat relying on an MSI Wind Intel Atom-based notebook, the total cost per station at $750 is typically less than most mainstream mobile devices and their requisite audio hardware. Given the convenient name, each station was assigned a letter of the alphabet, a corresponding wind name (e.g. Austru, Briza, Cyclone, etc.) and a static IP address. IP addresses are distributed evenly starting with 192.168.2.10 for Austru, *.12 for Briza, *.14 for Cyclone, etc. with the remaining odd IP addresses reserved for projected future growth.

Another advantage of such low-power setup is in the way it encourages creation of new content for the ensemble. The severe computational limits of individual computers in the ensemble literally force artists to approach new compositions by thinking about each station as a small piece of a much larger ecosystem, rather than composing essentially entire work on a single laptop and then "exploding" parts to individual machines. Based on the experience accrued through L2Ork, it is author's belief that the two approaches yield distinctly different results for performers and audience alike. Other advantages related to the choice of the aforesaid netbooks is reliable open-source driver support. As a result, L2Ork's setup supports all functions provided by the hardware, including standby, hibernate, and other advanced functionality that may be of use during pre-concert tech setup. For a more detailed breakdown of L2Ork's hardware infrastructure, please consult the NIME 2011 publication (I. Bukvic et al., 2010).

While initially based on modified Ubuntu 9.04 and 9.10 releases, L2Ork currently relies on the mainstream Ubuntu 10.04 LTS distribution. In the near term the ensemble is looking to migrate to a newer hardware with the next target OS choice likely being Ubuntu 12.04. With every Ubuntu upgrade, the author has identified fewer system modifications that were necessary to produce optimal configuration. The most notable alteration to the current iteration is the real-time kernel that enables even low-power Atom-based notebooks to deliver reliable low-latency performance. To further streamline system use in both rehearsal and performance scenarios the desktop has been enhanced with a set of operational and usability tweaks and customizations. Operational enhancements deal with use-specific needs that minimize potentially undesirable degradation in system's performance. These include real-time priorities for audio-related tasks, appropriate HD sleep and swappiness settings that abate potential xruns caused by HD usage, CPU throttling policy that automatically detects JACK server running and transitions in the “performance” mode, disabling screen savers and screen sleep timers that may interfere with the display during rehearsals and performances, associating static local IP address through the wired ethernet interface, and automatic reloading of bluetooth driver upon resuming the notebook to minimize potential problems with Wiimote pairing. In addition, usability enhancements include a series of shell script wrappers that provide turnkey initialization of various pieces stored in a form of application shortcuts (we will discuss these further below), as well as use of compiz (“Compiz Home,” n.d.) for the purpose of providing a more responsive and user-friendly desktop environment. A limited number of desktop shortcuts provides access to the most common functionality, like Web browsing, as well as a shortcut for force-quitting potential runaway processes. For the purpose of minimizing initial setup overhead, every system by default has auto-login enabled and the desktop's appearance is streamlined to promote seamless performer migration among different stations, should such a need arise.

### 3.1 Synchronization

L2Ork's infrastructure supports up to sixteen performers and despite its homogeneous design, several challenges arose from the ongoing attempts to synchronize stations and provide a new way to affect performance structure by cross-pollinating control data. A number of works written specifically for the ensemble rely on this component—for example in composition *Rain* one performer's action audibly percolates through other systems. Consequently, increased performer activity can quickly result in a relatively high amount of traffic to the point where the ensuing aural soundscape is built from stochastic percolations of short attack-based sounds propagated through the network. To achieve the said goal in this and other similar scenarios we rely primarily upon control data. Given the ensemble relies on wired, rather than wireless network, it is also possible to exchange high-bandwidth audio data among different members of the ensemble. In such, potentially high-network-traffic environment, wireless communication has proved surprisingly unreliable. Even when experimenting with industrial grade Cisco wireless router in conjunction with TCP packets over an isolated local network, we still encountered one or more machines receiving time-critical cues up to a second later. Therefore wireless communication has proven inadequate for any kind of synchronized music production.

The ensuing local area network is shielded from the external network traffic and thus limits potential network packet collisions. For this reason, all notebooks broadcast control data using UDP packets, enabling the individual stations to simply filter streams they wish to monitor and/or interact with. This has allowed the system to have a high flexibility requiring minimal configuration (Fig.1).

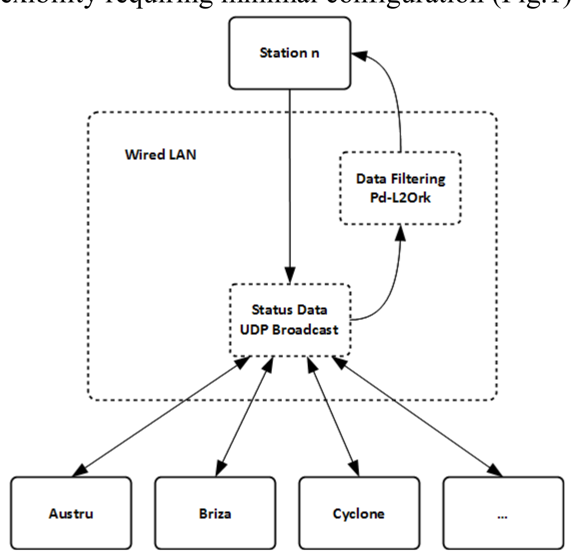


**Figure 1.** System synchronization.

Another challenge associated with synchronization is related to a seemingly trivial matter of connecting concurrently up to 16 co-located Nintendo Wiimotes. Discovering devices in such an environment simply isn't an option as there is no guarantee that the Wiimote will pair with the right station. For this reason, the system uses ethernet connectivity with predetermined local IP address as the foundation for all synchronization with external devices through the use of a series of shell scripts. All shell scripts are run as part of initialization sequence for each of the pieces. *whatismyip* script is in charge of detecting local machine's IP address that may be used for matching score part with the station/performer. *whatismywiimote* uses IP address to pair it with an appropriate Wiimote MAC address. The reason we did not rely on computer's MAC address as a means of identifying individual stations is because, unlike the hard-wired MAC address, the IP configuration can be easily altered. Therefore, through a simple alteration of the IP address, any station can be re-purposed to take over a role of a different computer in the ensemble. This has proved critical in situations where a potential hardware failure has rendered a particular station inoperable and yet where a work required first n contiguous machines, as is the case with the aforesaid *Rain* composition that echoes individual aural events through the ensemble.

Finally, as the ensemble grew, there was a growing overhead associated with patching and synchronization of each system, including latest L2Ork-related software updates, as well as revisions to compositions and the supporting software. This has proven a problem of exponential nature often requiring hours to administer on all 16 machines. It has also proven prone to human error due to its repetitive nature. Therefore, the ensemble developed a series of shell scripts (*l2ork-send* and *l2ork-do*) that akin to GRENDL technology (Beck, Jha, Ullmer, Branton, & Maddineni, 2010) provides near real-time synchronization with two notable differences:

1. The L2Ork synchronization system relies on the git framework, and
2. L2Ork is a homogeneous environment, which has made synchronization considerably easier (Fig.2).

As a result, at the beginning of every rehearsal, systems are synced from the instructor/developer's master machine in a matter of seconds. Additional supporting tools were provided using ssh and sftp protocols where instructor is capable of uploading new versions of system software that needs to be applied locally using administrator permissions. Such files are accompanied by an installer that administers all changes with minimal interaction from users. All these changes have vastly minimized the amount of time required to administer the ensemble, as well as provided for a more enjoyable experience among its users, many of whom have limited computer literacy and/or knowledge of music.

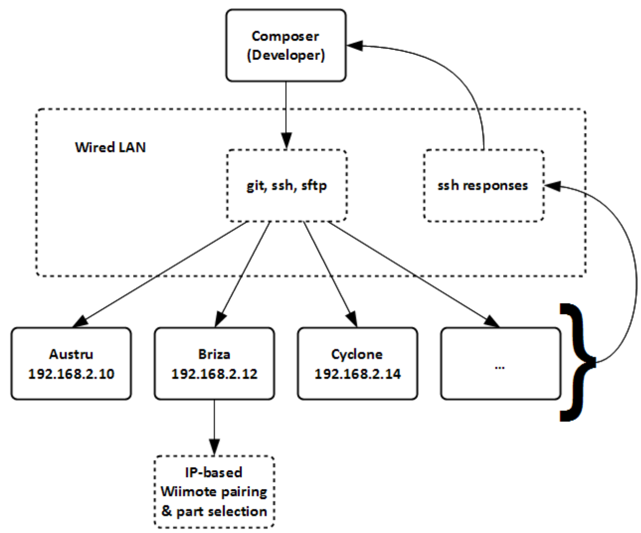


**Figure 2.** System synchronization.

### 3.2 Pd-L2Ork

For its audio-related digital signal processing and interfacing with external devices, L2Ork relies exclusively on the combination of JACK server (Davis & others, n.d.) and in-house version of Pure-Data (PD) (Puckette, 1996) titled Pd-L2Ork (Ivica Bukvic, n.d.). Former is integrated through QjackCtl JACK front-end that is also responsible for running a series of start-up scripts to ensure that the OS is running optimally (e.g. disabling wireless network to minimize potential confusion from having two active network connections, and setting CPU into performance mode).

Pd-L2Ork is an ambitious project in and of itself. Initially, the project's output was limited to upstream patches. In the fall of 2010, however, mainly due to increasingly divergent interests and needs between L2Ork and the upstream PD distribution, the author decided to create a fork. Since, the project has significantly altered the PD's core functionality with more than two hundred bug-fixes and new features. Based on the 0.42.6 branch of Pd-Extended ("Pd-extended — PD Community Site," n.d.), Pd-L2Ork focuses on two key areas: robust and stable operation of both the audio engine and GUI, and usability improvements to the runtime environment and editor. Unlike early iterations, more recent versions of Pd-L2Ork have had a crash-free streak for over a year, including over twenty evening-long shows and performances. The system's GUI is more resilient in high-bandwidth scenarios, while a series of new objects, including a threaded implementation of Linux-specific Nintendo Wiimote external that allows bi-directional communication without xruns have further enhanced its usability. The *disis_wiimote* external is of particular importance as L2Ork makes extensive use of haptic feedback to allow performers to monitor critical operation, even sense tempo and beat, without having to look at the screen, something that has proven particularly useful in practicing Taiji choreography.

Another notable area of Pd-L2Ork are extensive improvements to the editor. Some of them include improved scrolling algorithm, fixing all known GOP redrawing bugs and instabilities, addressing all known major bugs, adding infinite undo, ability for iemgui objects' and canvas' properties to be altered via GUI handles rather than through a separate properties window, and the re-skinning the GUI to give the application a more contemporary feel. Most of the core editing actions now rely on Tcl toolkit's "tags," e.g. displacing a large number of objects. This change has resulted in a significantly lower CPU footprint. Based on preliminary tests, moving a modest graph-on-parent-enabled abstraction now yields no noticeable CPU overhead on a low-power Intel i3 processor, while the same action using continuous redraws (legacy approach) encumbers up to 15% of the CPU on the same hardware. Objects can be also easily layered on top of each other and the *magicglass* signal monitoring tool has offered a more efficient approach to debugging. All of these features have helped streamline the production of performance interfaces (e.g. input/output monitoring and visual score engine) within PD's environment. Pd-L2Ork project is certainly interested in seeing its contributions adopted upstream. However, given the lack of control over this process or its pace, it is author's belief that maintaining a separate version has allowed the ensemble to dramatically increase the pace of software's development and as such remains the preferred choice.

## 4 Lessons Learned

In the world of computing there is a saying that the last 20% towards fine-tuning a turnkey system take 80% of the overall project time. Undoubtedly, the same holds true for L2Ork as well. Many of the seemingly trivial bug-fixes to the core Linux system and more notably pd-l2ork code base have taken weeks if not months to identify. Now, we are finally in a position where our integrated system is observed by a newcomer as something that just works. L2Ork infrastructure is therefore regarded as one fully integrated whole and the fact that it relies on Linux, beyond the obvious potential benefits of such an approach, are becoming entirely transparent to the user. It is author's conviction that this is where Linux's future lies. It is not the advocacy, or that curious yet often incomplete feature. It is the transparency coupled with unprecedented flexibility and power of such a system that make it a truly powerful platform. It is truly those last 20% that have consumed countless hours and yet rewarded in unforeseen ways.

Ironically, as we look forward to L2Ork's next rehearsal and a tour, the excitement behind the fact that the infrastructure is robust and stable has all but worn off. And even though some of us still remember the countless hours spent on many of pd-l2ork's novel features, we've finally arrived at the point where one can simply expect nothing less than stability, thinking this is simply the way it's supposed to be.

For additional information on the project, pd-l2ork and other supporting software, and video instructables, visit L2Ork at http://l2ork.music.vt.edu.

## 5 Acknowledgments

The author would like to hereby thank all L2Ork's Stakeholders and Sponsors without whose vision and support this project would've never seen the light of the day. Stakeholders are (listed in alphabetic order): Virginia Tech Alumni Relations, the Arts Initiative, College of Architecture and Urban Studies, Collaborative for Creative Technologies in the Arts and Design, Center for Human-Computer Interaction, Center for Instructional Design and Educational Research, College of Liberal Arts & Human Sciences, Creative Technologies Experiential Galleria, Digital Interactive Sound & Intermedia Studio, Institute for Creativity, Arts & Technology, Institute for Critical Technology and Applied Science, Institute for Distance and Distributed Learning, Institute for Society, Culture & Environment, Information Technology, Learning Technologies, Department of Music, Office of Academic Assessment, Outreach & International Affairs, School of Education, and the School of Performing Arts and Cinema. Sponsors are MSI Computer Inc., Renoise Inc., Roland Corporation, and Sweetwater Inc.

**References**

Beck, S. D., Jha, S., Ullmer, B., Branton, C., & Maddineni, S. (2010). GRENDL: grid enabled distribution and control for Laptop Orchestras. *ACM SIGGRAPH 2010 Posters* (p. 84).

Bukvic, I., Martin, T., Standley, E., & Matthews, M. (2010). Introducing L2Ork: Linux Laptop Orchestra. *Interfaces*, (Nime), 170–173.

Bukvic, Ivica. (n.d.). Virginia Tech Department of Music L2Ork - Software - Linux Laptop Orchestra. Retrieved February 8, 2012, from http://l2ork.music.vt.edu/main/?page_id=56

Bukvic, Ivica, Martin, T., & Matthews, M. (2011). Moving Beyond Academia Through Open Source. Solutions–Introducing L2Ork, Virginia Tech's Linux Laptop Orchestra. *Journal SEAMUS*.

Compiz Home. (n.d.). Retrieved February 28, 2012, from http://www.compiz.org/

Davis, P., & others. (n.d.). *JACK Audio Connection Kit (software)*.

Elliott, J. (2011, September 29). Virginia Tech's Linux Laptop Orchestra to perform at Virginia State Fair | Virginia Tech News | Virginia Tech. Retrieved February 28, 2012, from http://www.vtnews.vt.edu/articles/2011/09/092911-clahs-l2orkstatefair.html

International Association of Laptop Orchestras. (n.d.). Retrieved February 28, 2012, from http://ialo.org/doku.php/laptop_orchestras/orchestras

Kapur, A., Darling, M., Wiley, M., Vallis, O., Hochenbaum, J., Murphy, J., Diakopolus, D., et al. (2010). The Machine Orchestra. *Proceedings of the International Computer Music Conference* (pp. 554–558).

Pd-extended — PD Community Site. (n.d.). Retrieved February 8, 2012, from http://puredata.info/community/projects/software/pd-extended/?searchterm=pd-extended

Phillips, D. (2010, May). Introducing L20rk: the Linux Laptop Orchestra | Linux Journal. Retrieved February 28, 2012, from http://www.linuxjournal.com/article/10694

Puckette, M. (1996). Pure Data: another integrated computer music environment. *IN PROCEEDINGS, INTERNATIONAL COMPUTER MUSIC CONFERENCE*, 37—41.

Smallwood, S., Cook, P., & Trueman, D. (2009). Don't Forget the Laptop: A History of Hemispherical Speakers at 10. M. Kuo, D. Poxson, Y. Kim, F. Mont, J. Kim, E. Schubert, and S. Lin. *Proceedings of the New Instruments for Musical Expression*, Pittsburgh, PA, USA.

Trueman, D., Cook, P., Smallwood, S., & Wang, G. (2006). PLOrk: Princeton laptop orchestra, year 1. *Proceedings of the International Computer Music Conference* (pp. 443–450).

Wang, G., Bryan, N., Oh, J., & Hamilton, R. (2009). Stanford laptop orchestra (slork). *Proceedings of the International Computer Music Conference* (Vol. 505, p. 508).