The Integra Project

Jøran Rudi

NOTAM
Norwegian Center for Technology in
Music and the Arts, Oslo
joranru@notam02.no

Jamie Bullock

Music Technology department,
Birmingham Conservatoire,
Birmingham
jamie.bullock@bcu.ac.uk

Abstract

The paper describes the ideas for the Integra project, discusses the project in relation to similar initiatives, and describes the current development of commissions, performances and the supporting software *Integra Live*.

Introduction

The Integra project was initiated by Lamberto Coccioli at the Birmingham Conservatoire in order to address issues pertaining to preservation, composition and performance of contemporary, technology-dependent music. Composers of instrumental music that are not specialists in technology frequently find the demands of the software to be a barrier that discourages engagement with current techniques, and this corresponds well with reports from ensemble musicians that feel unease when confronted with technology for performance. This tendency is strengthened with the increased focus on interactivity, where the musicians' input is a determining factor for various signal processing routines executed by computers. These two challenges must be overcome in order to open up the field also for non-specialists, and there are several software initiatives that address them.

Archival and preservation issues are getting more attention as the heritage of technology-dependent music is growing, and it is sadly quite common for historically important and interesting works to disappear from concert bills, simply because the technology needed to have them performed is no longer easily available. These works can be less than twenty years old, and still have a place in contemporary memory. Several initiatives have been taken in order to address the concerns rising from this decimation of the repertoire, and to increase the re-performability. The approach taken in Integra has been to develop a software layer that the works will be migrated into; a software intended to make it possible to replace the underlying DSP engine when that might be needed, while retaining the music sounding as when it was composed, and with an identical interface.

The final issues addressed in the project were the contemporary music scene and its audience. Integra has commissioned a number of new works for ensembles and the *Integra Live* technology, and trained composers and musicians in this development of the repertoire. The chosen composers have a good standing in the European contemporary music scene, and the new works and performances have contributed significantly to the project. Summarized, the Integra project objectives are:

- To lower the threshold for use of technology for composers who combine composition for instruments with electroacoustic techniques.
- To commission new works for realization by the consortium and other ensembles.
- To bring key historical works of musical interest into performance-ready condition through migration into current technology, and have them re-performed.
- To engage and train composers and performers, and develop their technological skills.
- To perform a significant number of concerts throughout Europe.
- To develop international collaboration between research centers, ensembles and composers.

In sum, Integra is developing a new software environment for making music with live electronics, while modernizing works that use old technology, commissioning composers, and overseeing an extensive program of performances.

Related initiatives in preservation, migration and software development

Integra brings several issues regarding contemporary music creation and performance into the same project, and the discussion on previous initiatives will have to be divided into sections.

Preservation and migration

The focus on preserving cultural heritage has gained momentum during recent years, and has spawned efforts also in the musical domain. However, most preservation efforts have been directed towards preservation of physical musical scores, digitization and availability of the manuscripts through database searches, and those do not concern us here.

While migrating fixed-media works is a relatively trivial task, digital objects for live performance are fragile, and often depend on soft- and hardware with short life spans. There have been a few projects that have focused on resolving these issues. One example is MUSTICA (2003–4) [1], which concerned itself with the development of a database for archiving digital objects needed for performance of works with electronics. The project relied on the technical and musicological competence provided by GRM and Ircam, the two European partners in the project. The partners aimed for maintaining the data files in a manner that allowed for future interpretability, but the project did not secure performability, since data files were merely preserved, and not migrated into current soft- and hardware solutions or into a more generalized format.

The perhaps most important and ambitious project is CASPAR (2006-9) [2], and aimed to collect and preserve both scientific data and cultural content. The project succeeded in creating a good database structure for displaying the different elements of the technical requirements, but was limited to documenting and collecting the elements and presenting them in as authentic version as attainable. Thus, neither MUSTICA nor CASPAR realized direct performability of obsolete works, as implicated in Bonardi (2008) [3] and Esposito (2008) [4].

These two projects were followed by the Gamelan project (2009-12) [5]. The principal aims for Gamelan is to create a meta-environment for production of electroacoustic elements, making it possible to trace technical working methods in music composition with software, and formalizing, preserving and migrating this information to the meta-environment. In this manner, the project aims to make these methods available for reuse. A technical refinement is found in the ASTREE project (2009-11), described by Ciavarella (2009) [6] and on a on the Ircam website [7]. The project focuses its technical approach on recoding MaxMSP-patches (and externals) into generalized mathematical language for Faust [8], which can be compiled into the C programming language. At the stage of this writing (June 2011), the project has a proof-of-concept for limited implementation.

Software development for increasing usability

An important line of development of computer music software is from the text-based music-N environments to graphics-driven user interfaces, and as the complexity in methods for signal processing and other musical organization has grown, so has the complexity of professional software. While plug-and-play software is sufficient for many musical expressions, a much-used paradigm in the computer- and classic electroacoustic music community is module-based patching systems such as MaxMSP, PD [9] and KYMA [10], where the user builds up her own subprograms (patches) by the combination of smaller routines or subprograms (objects or other patches).

With this flexibility for the composer comes challenges for the musician – every patch will work differently, have different functions, look different - and leave the user to navigate what is often not altogether intuitively organized collections of objects. For the non-specialist performer, this contributes to raising the threshold for use of technology. A need for simplification has been recognized for both composers and musicians, however, it is unattractive to give up the opportunities that the DSP engines provide for complex and demanding processing. The development of graphical elements for Supercollider is a step in this direction, as is Jamoma [11], which in essence is an extended graphical front end to MaxMSP.

With the rapid development of modular software follows a need for open standards for coding and message transfer between applications and environments, and several formats have been proposed. Open Sound Control (OSC) was originally intended to share music data between instruments and computers, as a more powerful alternative to the more limited MIDI-protocol. XML is a more general approach to formatting documents for machine-readability. Both standards are in much use.

The Integra approach and challenge

The basic idea at the project start-up in 2005 was to develop the technology and have it be populated with objects created through the new commissions and the migrations of the existing works. The software was going to be used to support both composers and musicians in the creation and performance of commissioned and migrated works. Integra commissions and works to be migrated were to be selected based on the repertoire development of the ensembles in the consortium. The tasks of supporting composers and musicians, and of migrating works from obsolete technology were to be distributed between the research centers, and attention to regional cross-fertilization by the combination of ensembles, composers and ensembles from across Europe was important.

Composers would visit the research centers for technical development, and musicians from all ensembles were to be hosted by the research centers for workshops. In this manner, composer- and musician training was to be distributed across Europe, so that ensembles and research centers would expand their network, and gain insight into working methods and practices from several regions. For example, the Italian composer Andrea Cera wrote for the Paris ensemble Court-circuit, and had NOTAM from Oslo as a supporting centre. In addition to these structured efforts at the core of the project, Integra would invite musicians, composers and administrators from different environments across Europe to observe and be part of several concerts, workshops, and so on.

The complexity of developing a model for, and actually building the software, as well as the high degree of interaction between the different parts of the project, posed serious challenges to the project. Also, the funds allocated for technical development were insufficient in comparison to the task at hand. As a consequence, the technical results were less significant than planned for. During the first part of the project, a prototype was developed, and it was only during the second part of the project that the Integra software became operational as a usable tool for composition and performance. Following from this, the commissions scheduled for performances during the first three years of Integra could not be fully realized within the Integra software, but were developed using existing tools such as MaxMSP and PD. The 17 works that were migrated during the first part of the project were nonetheless modernized and brought back into performable state.

Commissions, migrations, training, performances

One of Integra's goals was to further develop the repertoire of contemporary music where acoustic instruments and ensembles also employ technology. In all, Integra has commissioned 16 works [12], all from composers with an international profile. Commissions from the first part of the project were performed 22 times during the first part of the project, integrated into tours, concert series', and festival programs. As described above, it was only during the second part of the project that the software was developed, and the 5 new commissions are all being written using the software, thus developing it further through the collaborations between composers and research centers. There were 11 training sessions for composers during the first iteration of Integra, and there will be approximately the same number during the second iteration, bringing the number up to ca. 20.

At the close of the project, 33 works will have been migrated [13]. The migrations in the second part of Integra are being executed using Integra tools, and for NOTAM's part, we will re-migrate our previous migrations to make them Integra-compliant, and thus extend the library of functions according to the original plan. During the first iteration, migrated works were performed at 13 events.

The ensembles that have been part of Integra during one or both of the project iterations are:

> Bit 20 ensemble (Bergen), Ars Nova (Malmö), Athelas Sinfonietta (Copenhagen), Grup Instrumental (Valencia), Ensemble Court-circuit (Paris), and Birmingham Contemporary Music Group (Birmingham).

Musicians from these ensembles have been taking part in training sessions at the different centers, receiving assistance in learning migrated or commissioned works, and during the first iteration, there were 10 training sessions.

The project has sought collaboration with existing festivals, such as the Ultima festival in Oslo and the Festival International de Música Contemporánea in Valencia, for performances.

The most important manifestation of the project was the Integra festival and conference in Birmingham in 2008, which will be joined by the Integra festival that is planned for Copenhagen in November 2011.

Integra Tools

The key of the Integra project is the software tool *Integra Live* [14] for composition and performance of music with live electronics. In keeping with the original aim of lowering the threshold for accessibility, much emphasis has been placed on the user interface, which is built on a model proposed by researchers from McGill University. In the GUI, users can choose between an arrange and a live view. The arrange view allows for moving and connecting modules, and a selected module will always display its parameters and parameter values. Live view is tailored for the concert situation, and selecting regions is facilitated for ease of rehearsal and repetition. The signal processing takes place in modules that can be selected from a list, and presets are set so that modules immediately work, much in the same manner as in Kyma [15], another DSP and composition environment. Modules can be patched together on the fly, and dropped into the signal processing chain in a completely flexible manner.

The current technical partners in Integra are Birmingham Conservatoire, CIRMMT (McGill University), Institut für Elektronische Musik und Akustik, Muzyka Centrum, Malmö Academy of Music, and NOTAM. During the first part of the project, the group also included La Kitchen, Krakow Academy of Music, Lithuanian Academy of Music and Theatre, and for a short period also SARC. The principal developers have been Jamie Bullock, Henrik Frisk, Thomas Musil, Leighton Hargreaves and Kjetil Matheussen.

At the onset of Integra, there were no existing preservation initiatives that aimed for making obsolete works performable; the ongoing projects were merely collecting scores and other performance materials. At the Integra kick-off in Oslo in 2005, it was decided that the Integra tools should make it possible to migrate works into a format that would be forward-compatible, by introducing a general descriptive layer, and by using a general, open standard for communication between this layer and the DSP engine [16]. In this manner, it was thought that it should be easy to replace the DSP engine by new engines, should they arrive in the future. This layer was implemented as an OSC namespace to facilitate inter-module communication. Early tests at NOTAM, as well as a study by Bullock (2006) [17], showed that placing this extra OSC layer as a wrapper around individual DSP modules added unnecessary message processing overhead, causing increased CPU loads even with DSP functions of low complexity. The model for the protocol was therefore revised so that whilst individual modules used native messaging to communicate, an abstraction layer was introduced to present a standard OSC interface to external software and hardware devices. This abstraction layer was implemented as a shared library (libIntegra), which handles module intercommunication, state saving and other common tasks.

libIntegra is capable of reading and writing module state (including the module connection graph) in a newly devised IXD (Integra extensible Data) format [18], which is based on the XML standard and can be validated via an XML schema. This separation of module interface, state storage, and DSP implementation allows for the DSP backend to be changed whilst preserving a common interface and data storage format. Using the module meta-data pulled from the Integra database at runtime libIntegra is able to provide the Integra GUI with all the information it needs to automatically build module listings and enumerate module controls.

Such information includes module parameter ranges, units and scaling. Additionally libIntegra dynamically generates an OSC address for each module that is instantiated in the DSP host. For example if we instantiate a SpectralDelay module, it's inLevel parameter may be addressed using '/myproject/SpectralDelay1/inLevel'.

The graphical user interface (GUI) for Integra Live acts as a client to libIntegra, using the libIntegra XMLRPC interface for communications. XMLRPC was chosen as a simple standard, which unlike OSC, is intended for bi-directional communication. This bi-directionality is necessary to keep the GUI state in sync with the state of the libIntegra and the DSP host. The GUI is written in Adobe Flex and acts as a 'dumb client' to libIntegra, using the XMLRPC API available at http://www.integralive.org/incoming/api.html. For example, the GUI must call command.new() on the server, passing in the module name, instance name and parent path as arguments to create a new module. This message is passed to libIntegra, which passes an instantiation command to the DSP host via a host-specific bridge.

Due to time constraints, the initial release of Integra Live has been for Mac only, however builds have been made for Windows and GNU/Linux, with minimal changes to the codebase, and once these builds have been tested and installers have been made, additional releases will be made. Also due to time limitations, Integra Live currently only supports Pure Data as a DSP host. The project svn repository contains a working SDK for developing Integra modules in Max, but currently an insufficient number of Max module implementations exist to create a working system. This could be seen as a limitation in the scalability of the system since every Pd module must have an equivalent working in other supported hosts (e.g. Max) in order for Integra projects to be portable between different DSP hosts. A more sustainable solution is to create an abstract representation of the core module implementation and auto-generate host-specific implementations from this. Work is currently underway at NOTAM to specify Integra modules using FAUST functional notation, which solves this sustainability problem.

One of the advantages of using an open environment like Pure Data as a DSP host is that new modules for the Integra Framework can be developed by anyone with a knowledge of Pd. Integra seeks to make it easy-as-possible to develop new modules by providing a cohesive SDK with a range of templates available for standard module configurations such as 'stereo in – stereo out' or 'mono out only'. To make an Integra module, the module developer must define their module's interface in the Integra database by filling in a series of simple forms. Then creation of a module is then as simple as opening up an existing template (Pd patch) and adding a [route] object with the names of the module attributes listed. However, despite this apparent simplicity, community uptake in contributing modules to Integra has been slow. This could be explained by a lack of documentation and the fact that the module development SDK is not officially released. The intention is to eventually make a screencast for the project website explaining the module development process, along with a link to the SDK download on the main downloads page.

Integra Live is an order-of-magnitude easier to use than current tools for live electronics such as MaxMSP and Pd. It takes months for a new user to get musically interesting results from Max, whereas this can be achieved in minutes with Integra Live. However, Integra Live lacks the flexibility of programming environments like MaxMSP and Pd, and so existing users of those environments are reluctant to change tools. Additionally, new users are sometimes frustrated by the limitations of the software.

Integra Live was released spring 2011, and version 1.0.9 contains a number of objects [19] that have been developed in conjunction with composition or migration of works.

Integra education

In order to introduce the software, all composers have been working with the technical partners in the project, however, the software *Integra Live* became useable only during the late winter 2010/2011. The educational aspects of the program has suffered somewhat because of this delay, but all composers have worked with technical partners, and with few exceptions received assistance in their compositions. The commissions in Integra 2 will all be realized using *Integra Live*, and as a result of this, the software will be further developed and add objects to the list mentioned above.

The training of musicians have followed much of the same idea, musicians from all ensembles have been, or will visit one of the technical partners in order to work on the migrated and commissioned pieces, and the project has resulted in a series of successful collaborations.

In additions to these events planned for the project members, there have been 3 national workshops [20], led by the research centers IEM, NOTAM and Birmingham Conservatoire for the interested professional and voluntary community, where the software has been introduced and demonstrated.

Summary of results, future work

Given the number of concerts, and the attention the public activities have received, it is fair to say that the commissions, the training of composers and musicians, and the concerts have been successful on the scale normally expected for contemporary music in Europe. The migration of historical works must be said to have yielded welcome additions to the repertoire.

The original idea for the technical development exceeded the capacity of the consortium, and was in essence pared down to the development of an easy-to-use GUI with a number of the most common tools for composers coded and appearing as finished objects. The software presents a front end to PD, and the GUI's ease of use has moved the project a good step towards achieving its goals of lowering the threshold for non-specialists. The project has not managed to implement forward-compatability in the originally planned sense, and the future of *Integra Live* now remains with the number of users that will continue to develop modules and add them into the program, as well with the success of the pieces that are relying on *Integra Live* for their realization.

Acknowledgements

Integra Live was developed in order to meet the objectives of Integra, a 3 year EU-funded project led by Birmingham Conservatoire, following another 3-year EU-funded project, "Integra, A European Composition and Performance Environment for Sharing Live Music Technologies", both part-funded by the Culture program of the European Commission.

References

[1] BACHIMONT Bruno *et. al.*, "Preserving Interactive Digital Music: A Report on the Mustica Research Initiative", in *Proceedings of the Third international conference on Web delivering of Music (Wedelmusic),* Leeds (UK), 2003.

[2] http://www.casparpreserves.eu/ (visited March 9, 2011).

[3] BONARDI Alain *et al.* "Will Software Modules for Performing Arts be Sustainable ?", in *Proceedings from The 14th IEEE Mediterranean Electrotechnical Conference* (MELECON'08)*,* Ajaccio (France), 2008, actes sur CD-ROM.

[4] ESPOSITO Nicolas and GESLIN Yann, "Long-term Preservation of Acousmatic Works: Toward a Generic Model of Description", in *Proceedings from The 14th IEEE Mediterranean Electrotechnical Conference* (MELECON'08)*,* Ajaccio (France), 2008, actes sur CD-ROM.

[5] http://www.ircam.fr/309.html?tx_ircamprojects_pi1%5BshowUid%5D=55&tx_ircamprojects_pi1%5BpType%5D=p&cHash=c2f256f503&L=1 (visited March 9, 2011).

[6] http://www.rinascimento-digitale.it/eventi/conference2009/slides16-2/ciavarella.pdf (visited March 9, 2011).

[7] http://www.ircam.fr/59.html?L=%5C%5C%5C%5C%5C%5C%5C&tx_ircam_pi1%5BshowUid%5D=46&ext=1&L=1 (visited March 8, 2011).

[8] BARTHELEMY Jérôme *et al.*, "First steps towards an organology of virtual instruments in computer music", in *Proceedings from International Computer Music Conference*, 2010, New York (USA), pp. 369-372.

[9] http://puredata.info/ (visited March 9, 2011).

[10] http://www.symbolicsound.com/cgi-bin/bin/view/Products/WebHome (visited March 9, 2011).

[11] http://www.jamoma.org (visited March 9, 2011).

[12] Composers that have been commissioned by the Integra project are: Ondrej Adamek (Ça Tourne Ça Bloque), Natasha Barrett (Crack), Ed Bennett (Noise Machine), Malin Bång (Sparkling Box), Andrea Cera (Dueling Zombies), Alvin Curran (Circus Maximus), Tansy Davies (Grind Show), Juste Janulyte (Eclissi), Pierre Jodlowski (Respire), Mauro Lanza (new work), Philippe Leroux (Apocalypsis), Hilda Paredes (Revelación), Alwynne Pritchard (Frame), Iris ter Schiphorst (new work), Michele Tadini (Walking Through Boundaries), Rolf Wallin (Strange News).

[13] The following works have been/will be migrated: Pierre Boulez (*Dialogue de l'ombre double*), Roland Dahinden (*Lichtweiss*), Ivan Fedele (*Elettra*), Luca Fransesconi (*Lips*, *Eyes*, *Bang*), Gerard Grisey (*Prologue*), Jonathan Harvey (*Madonna of Winter and Spring*, *Ricercare*, *Valley of Aosta*, *Wheel of Emptiness*, *White as Jasmine*), Philippe Hurel (*Diamants Cycle*, *Fragments de Lune*), Krzysztof Knittel (*Points and Lines*), Wlodzimierz Kotonski (*Spring Music*), Miklos Marós (*Manipulation III*), Tristan Murail (*Treize Couleurs Du Soleil Couchant*, *Winter Fragments*), Thea Musgrave (*Narcissus*), Luigi Nono (*Omaggio a György Kurtág*), Arne Nordheim (*Partita Für Paul*), Kent Olofsson (*Tarpeian Rock*), John Rea (*Treppen Music*), Roger Reynolds (*Traces*, *Ping*, *Transfigured Wind II, III and IV*), Edwin Roxburgh (*At the still point of the turning world*), Kaija Saariaho (*Lichtbogen*), Asbjørn Schaathun (*Our Whisper Woke No Clocks*, *Physis*, *'S'*), Lasse Thoresen (*Abuno*).

[14] BULLOCK Jamie, Daniel BEATTIE and Jerome TURNER, "Integra Live: A new graphical user interface for live electronic music", in *Proceedings of the New Interfaces for Musical Expression (NIME)*, Oslo (Norway), 2011, pp. 387-392.

[15] http://www.symbolicsound.com/cgi-bin/bin/view/Products/WebHome (visited March 9, 2011).

[16] BULLOCK Jamie and Henrik FRISK, "Libintegra - A system for software-independent multimedia module description and storage", in *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen (Denmark), 2007, pp. 43-46.

[17] BULLOCK Jamie, *The Integra protocol: a feasibility study,* 2006. Available at: http://www.integralive.org/community/ (visited May 24, 2011).

[18] BULLOCK Jamie and Henrik FRISK, "Libintegra - A system for software-independent multimedia module description and storage", in *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen (Denmark), 2007, pp. 43-46.

[19] Integra Live modules in version 1.0.9: 4-1 mixed, Audio in/out, Bandpass filer, delay, dirstortion, envelope follower, flanger, granular delay, hipass filter, limiter, long tap delay, lowpass filter, material simulator, MIDI control in, notch filter, octo sound file player, octo soundfile player, onset detector, percussive onset detector, phase vocoder, phaser, piano reverb MSP, piano reverb strings, ping-pong delay, pitch detector, pitch shifter, quad audio in, quad audio out, quad auto panner, quad granular synthesizer, quad panner, quad X-Y panner, resonant bandpass filter, resonant lowpass filter, reverb, ring modulator, scalar, short tap delay, simple sampler, slicer, soundfile player, spectral delay, spectral vocoder, stereo audio in, stereo audio out, stereo granular syntehsizer, stereo panner, stereo reverb, stereo soundfile player, stereo reverb, tap delay, test source, vibrato or chorus.

[20] Valencia May 13, 2011, Oslo May 28, 2011.