Interactive multimedia systems for engineering education in acoustics, synthesis and signal processing

ANDY HUNT[†*], DAVID M. HOWARD[†], ROSS KIRK[†], KINGSLEY ASH[†] and ANDY M. TYRRELL[†]

This paper describes programmable multimedia systems, developed at the University of York, which are used extensively for teaching on a variety of music technology and mainstream engineering courses. Software and hardware systems are described for the physical modelling of acoustic spaces, and for constructing interactive synthesis and signal processing networks. Details are given on how these have been successfully integrated into higher education programmes at York.

1. Introduction and background

Engineering is a creative art form. Though it is primarily viewed as a science, and much emphasis is placed on analytical techniques, an engineer is basically a person who *creates* technological solutions for human problems. This spark of creativity in engineering is perhaps often overlooked by those who teach purely analytical methods. Many students find that they can follow the logic of a 'worked solution' in the lecture theatre, but then are completely bemused when faced with a blank sheet of paper and a problem to solve. In this respect they are in a similar position to a musical composer who is faced with a similar blank sheet of paper at the commencement of writing a piece of music. In order to proceed with the creative task, both the engineer and the composer need not only a knowledge of the 'building blocks' involved in their craft, but also a good deal of imagination, experimentation and experience in constructing these bricks into the final product.

It was this acknowledgement of the *complementarity* of music and engineering that led us, at York, to create the first postgraduate course in Music Technology in 1986. Since then the course has flourished, and has led to undergraduate courses here and at several other universities. The course ethos is that separate streams of engineering-trained and music-trained students can come together to form a fruitful and successful collaboration (Hunt and Kirk 1997). Indeed, both sets of students find that they can learn *from each other*. Whilst the music students pick up engineering terminology, logical thinking and formula manipulation, the engineering students find they are learning about creative thinking, aesthetic judgement and human communication.

Details of the course can be found on the web (first web reference), but it is useful to outline the structure here as it will help to explain why we have developed

[†] Electronics Department, University of York, Heslington, York YO10 5DD, UK.
[*] To whom correspondence should be addressed. e-mail: adh@ohm.york.ac.uk

European Journal of Engineering Education
ISSN 0343-3797 print/ISSN 1469-5898 online © 2001 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/0304379011003357 4

our own engineering technology support systems. The course involves four basic sections: a series of *core* taught courses; a choice of special interest groups based on research topics; a software engineering project; and a solo project.

The core courses are: digital audio processing, human perception of sound, electronic musical instruments, signal preservation and aesthetics. These cover the range of issues from why and how humans compose and perform music, through to the technological implementation of hardware and software systems for music composition and performance. The special interest groups offer the students a chance to interact with the staff and each other in a range of research topics. These change from year to year, but include multiple media installations, composition, 3D sound spatialization, the human singing voice and special needs technology. Assignments involve different degrees of group work. Students learn the art and dynamics of working in teams on creative exercises. A special unit on software engineering tests these abilities to the full, as teams of students form companies with quality assurance procedures, with the aim of designing and producing a large-scale piece of audio processing software. Our graduates often refer to this group work, with hindsight, as the most valuable aspect of the course. A solo project is then carried out by the student, and usually involves a large-scale software, hardware or composition project, to include an element of innovation.

Throughout the course, students are actively encouraged to experiment with the building blocks of music and engineering technology on systems that have been designed and built at York. Since they are produced in-house, the students then have the opportunity of getting involved with the furtherance of such systems at a research and implementation level. This paper explains how some of these systems were developed, and suggests how they have been used to enhance the engineering educational experience.

2. Software systems for training in acoustics

2.1. *Introduction*

The teaching of basic acoustic principles demands the ability to think on an extremely small scale in discussions relating to sound propagation, whilst attempting to imagine its associated dynamic properties. Sound is so often simply taken for granted and considered as being somehow intangible and therefore difficult to work on, appreciate and understand. The main aims of teaching acoustics include describing how sound propagates, as well as its nature and behaviour in different acoustic contexts (e.g., Howard and Angus 2000). A key obstacle to achieving these aims is the difficulty of any form of direct observation of acoustic pressure disturbance and the consequent difficulty of appreciating the fundamental principles involved.

One approach that has been adopted in the past involves the use of a shallow tank with a plate glass bottom and approximately 1 cm of water in it. This tank is placed over a light source and above the tank is a mirror to reflect the light that has passed through the water on to a screen. This apparatus is known as a ripple tank and it enables a number of properties of wave motion to be illustrated. Clear photographic illustrations can be found in Bragg (1920), where an account is given of its use in Sir William Bragg's lectures on sound to the Royal Institution in London. Disadvantages of this apparatus include its size, the presence of water and the fact that the sound waves being illustrated cannot be heard acoustically.

Recent research advances in the Music Technology Research Group at the University of York in the field of physical modelling enable acoustic wave propagation to be simulated accurately on a computer. The wavefronts can be observed graphically and the sound at any position in the space can be played via a loudspeaker. The software available for teaching acoustic principles has been developed as a spin-off from research work on physical modelling for: (a) sound synthesis (Pearson and Howard 1995, 1996), (b) two-dimensional (Murphy and Howard 1998, 1999a, b) and (c) three-dimensional (Campos and Howard 2000) wave propagation to simulate the acoustic properties of enclosed spaces.

This section describes the basic principles underlying these areas of physical modelling and the application of the software for teaching acoustics to undergraduate students reading for BEng and MEng degrees in Electronic Engineering with Music Technology Systems as well as postgraduates reading for MA and MSc degrees in Music Technology at the University of York.

2.2. *Computer physical modelling for acoustics teaching*

The computer simulation used for teaching acoustics is designed to function as a one- or a two-dimensional point mass and spring equivalent model. It takes its origin from the one- and two-dimensional physical modelling music synthesis system described by Pearson and Howard (1995, 1996) which enables strings and membranes, respectively, to be simulated, as well as the two-dimensional room acoustic simulation system described by Murphy and Howard (1998, 1999a, b). It therefore enables strings and membranes to be simulated.

Any number of strings and/or membranes can be incorporated into a given simulation and these can be connected at arbitrary points to produce either a virtual instrument of any configuration or a two-dimensional simulation of acoustic pressure propagation in an enclosed space. The masses of the point masses and the tension of the springs can be set separately for each element (string or membrane) of a virtual instrument. The boundaries can be firmly fixed, left free or have a damping gain factor between 0.0 and 1.0 applied. One of the following acoustic excitations can be applied to any single point mass anywhere in the simulation on a string or a sheet:

- pluck
- bow
- sinewave
- squarewave
- random noise
- live from a microphone.

Any degree of damping can be applied to any point mass to simulate the absorption of acoustic energy. A digital audio output is obtained from any number of virtual microphones placed at user-designated point masses, where placement is typically at one (monophonic output) or two (stereophonic output) masses.

2.3. *Example acoustic simulations*

The system finds application in teaching basic acoustic principles as a means of demonstrating wave propagation visually and acoustically. In our postgraduate teaching, students are able to access the system to enable support outside formal teaching sessions via a network of Silicon Graphics machines connected to an eight-node Silicon Graphics Origin 2000 machine.

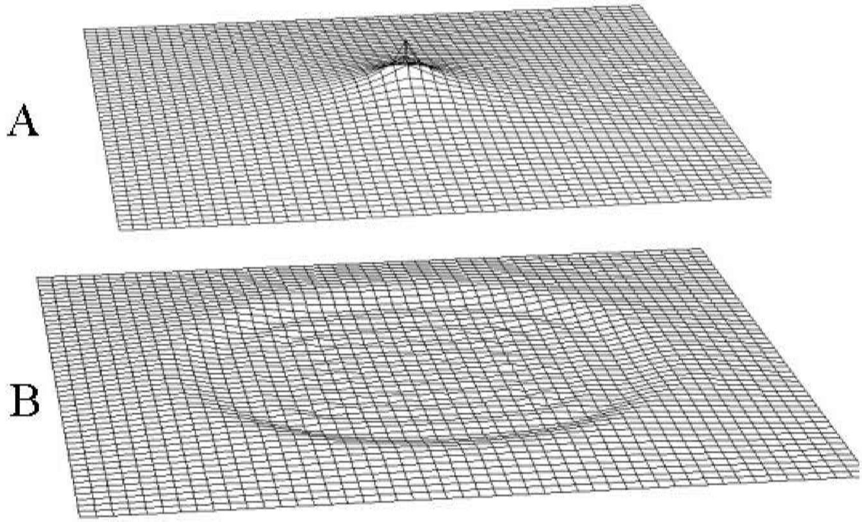

Figure 1. Graphic snapshots from a physical modelling simulation of a sheet plucked in the centre.

Acoustic simulation is generally introduced as a single one-dimensional structure simulating a string, thereby providing a comparatively simple acoustic system to be studied. The propagation of energy along the string can be shown and the nature of reflections at fixed ends can be illustrated and discussed as the basis for understanding what happens in stringed instruments as well as pipes.

Of rather greater visual interest is the two-dimensional display, which models acoustic propagation in two dimensions, wave motion away from a source, how it is reflected from boundaries and the effect that obstacles in its path have. Figure 1 shows a square sheet and a pluck (acoustic impulse) excitation located at the centre of the sheet, as shown in figure 1(A). The energy propagates outwards from the point source pluck equally in all directions, and the propagation of a circular wavefront can be seen in figure 1(B). This simulation provides an illustration that is very similar to that obtained from a ripple tank involving, as it does, transverse wave motion. (This arises as a consequence of the method by which the model is implemented. Each point mass is constrained to move within a vertical slider at right angles to the direction of wave propagation.) In order to provide illustrations of wave motion that more appropriately illustrates longitudinal wave propagation of acoustic pressure waves, thereby reminding students of the nature of the acoustic wave itself, an alternative graphical representation is available where the amplitude of excursion of each point mass is indicated as the darkness of marking on a grey scale plot. This plot shows areas of vertical point mass movement upwards (which could be considered to represent longitudinal wave compression) in dark shades of grey and areas of vertical point mass movement downwards (which could be considered to represent rarefaction) in lighter shades.

Figure 2 shows two plots of standing wave patterns in a two-dimensional square sheet. This can be used to demonstrate the standing wave nature of the modal region of a room's acoustic behaviour. In the figure, the boundaries are fixed and the

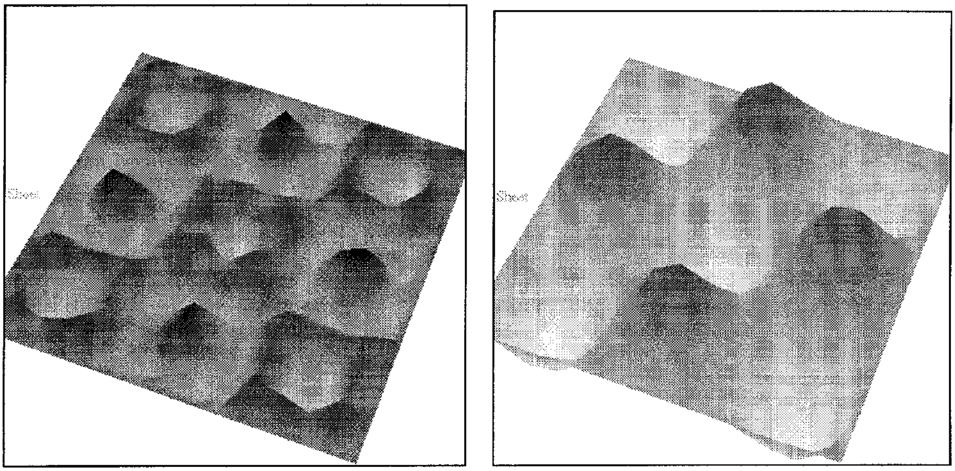

Figure 2. Graphic snapshots of standing wave modal resonances in a square sheet (2nd mode left; 3rd mode right).

excitation is sinusoidal at the mid point of the sheet provided from an external oscillator. In the left-hand plot of the figure, the frequency is set to that of the second standing wave mode along each axis, which is the same along each axis because the sheet in this example is square. The velocity nodes (no movement) and antinodes (maximum movement) are clearly revealed. Since an external sinewave oscillator is employed in practice, the modal behaviour can be explored at different frequencies by changing the input frequency with the model running in real-time. In the right-hand plot of figure 2, the third standing wave modes are shown where the frequency of the sinusoidal excitation has been increased by a factor of 3/2.

The two-dimensional sheet model also enables the effects of obstacles to be investigated and illustrated and the effect of placing a small opening in the path of an acoustic wavefront is illustrated in figure 3. A two-dimensional sheet has all of its boundary point masses fixed. In addition, those along its mid-line are also fixed, except for the centre 5 which form a slit. This could, for example, be modelling the effect of an open door between two rooms. The frequency of a sinusoidal input has been chosen such that the wave propagates through the slit and spreads out on the other side; an effect known as *diffraction.* The spreading extent of the diffraction depends on the wavelength of the sound wave relative to the width of the slit. The wave is diffracted strongly if the wavelength is large compared to the slit width as illustrated in the figure, and not at all (the wavefront continues in a straight line beyond the slit) when the wavelength is very much smaller than the slit width.

3. Software systems for sound and image synthesis

3.1. *Introduction*

In this section we describe how the students use a flexible software package to construct networks of digital signal processing building blocks. The resultant

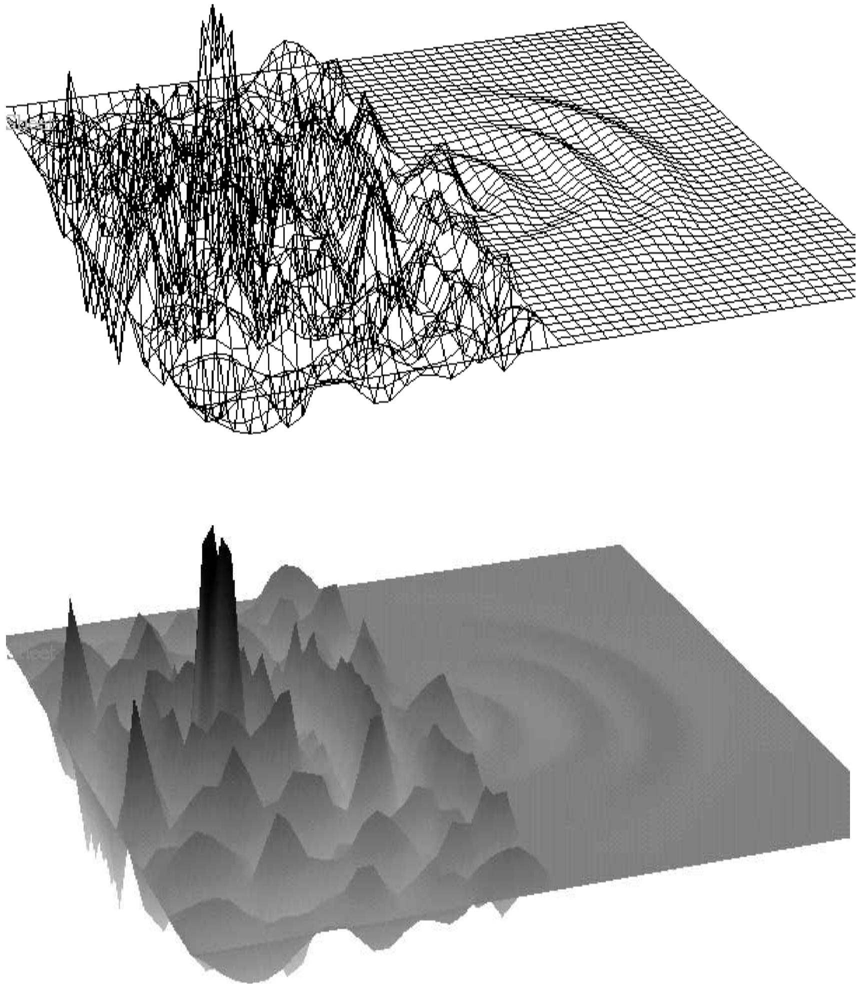

Figure 3. Grid (upper) and grey scale (lower) graphic snapshots illustrating diffraction.

systems produced by the students take the form of musical instruments or even multiple-media installations, but actually consist of complex webs of signal processing functions. Thus, the engineering principles are given a creative artistic outlet, which in turn improves the creativity of the engineers.

3.2. *The MIDAS audio-visual toolkit*

MIDAS (Kirk and Hunt 1996) is an acronym for the Musical Instrument Digital Array Signal processor. It was conceived in the early 1990s as a test-bed for real-time performance control of audio signal processing algorithms and for associated studies in human–computer interaction. It allows the construction of user interface experiments with real-time synthesis systems, and has since been under development at the University of York.

MIDAS allows users to manipulate a 'toolkit' of audio-visual algorithms for

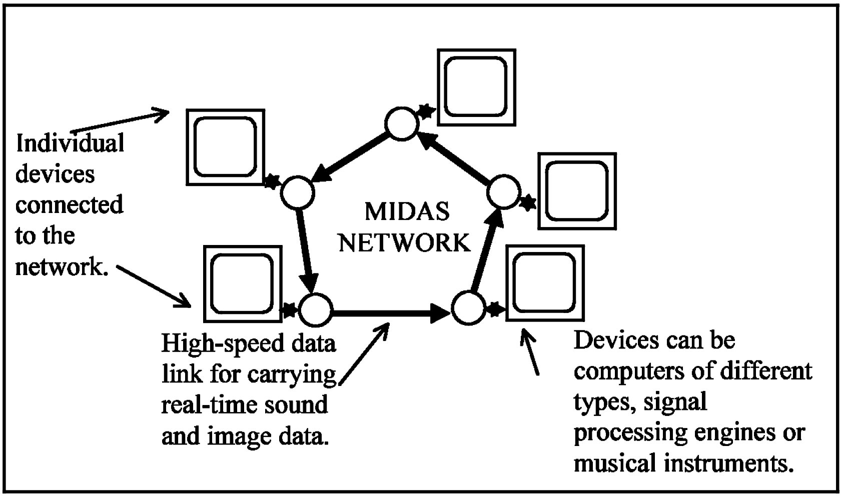

Figure 4. The MIDAS system.

constructing interactive systems. It also provides a means of distributing these algorithms over a network of connected computers of different types, so as to maximize the available processing power. Full details of the system, as well as how to download it free of charge, are available on the MIDAS web site (second web reference). A summary of its key features is now provided so that those unfamiliar with the system will understand how it has been developed and used as part of the engineering education philosophy at York.

MIDAS is based around the concept of the *Unit Generator Process (UGP)*. A UGP is a piece of code that handles an individual task, such as drawing a rectangle, establishing the current position of the mouse or generating a sound output. The concept will be familiar to computer musicians in the form of audio units, such as oscillators and filters in programs such as *Csound* (third web reference), but MIDAS UGPs are designed to run in real-time on distributed processors, and can deal with generalized signal information (including graphics). UGPs can be connected into a *network* which performs a more complex audio-visual processing task. MIDAS is therefore capable of being configured into performing any possible synthesis or processing method.

Figure 4 is a graphical representation of a MIDAS network running on several different computing platforms.

It is helpful for students if the MIDAS UGPs are divided into a series of functional categories for easy reference, such as Audio, Graphical & User Interface, Maths & Logical, Musical Instrument Digital Interface (MIDI) Handling and Signal Manipulation.

The benefits of producing the MIDAS system have been manifold. Not only do we now have a potentially cross-platform architecture for audio-visual interaction, but the code is also easily accessible. It has allowed practical exercises in multimedia, MIDI systems and audio synthesis and control to be run on the Masters course at York. We now describe two particular instances of the use of MIDAS within an engineering educational context.

3.3. *Electronic musical instruments*

The electronic musical instruments course unit encourages the students to investigate the nature of musical instruments, and then to use this knowledge to design and build (in teams) a new electronic musical instrument. The assignment involves specifying, then implementing:

- a novel *interface*
- a software-based sound synthesizer
- the connection between the two.

The MIDAS system (outlined above) is used to create the sound synthesis engine. Students experiment with networks of audio unit generators which will make their required sound. They are initially provided with a tutorial series of ready-made network examples that they can run in order to learn how the network configuration affects the resultant sound.

The examples progress from a single-tone generator, via a series of additive synthesis and frequency modulation examples, to networks which contain an element of sequencing (notes which repeat automatically). Once the students have produced a network that makes sound, they then need to connect it to their newly designed playing interface. This interface is a hardware construction, built out of assorted materials, and including sensors to transduce the musician's movements into electrical signals. These signals have to be entered into the computer and routed to the MIDAS sound algorithm. This is done via a MIDI connection (e.g., Hunt and Kirk 1999). MIDI provides a serial interface for coding and transmitting musically-based human control gestures (fourth web reference).

The first stage of this conversion process (from human gesture to MIDI signals) is carried out by a piece of hardware, designed at the University of York, called *MidiCreator*. This accepts up to 16 sensors whose voltage or resistance can vary, and converts these into MIDI messages by means of an on-board mapping program. This program can be set up in advance on a PC and downloaded on to a smart card. The system can then be reconfigured during use by changing the smart card. Full details of the *MidiCreator* system can be found on its web site (fifth web reference).

The final stage of the process is to accept the MIDI signals into the computer, and then configure the MIDAS sound-synthesis network so that it can receive data from the incoming MIDI messages. The whole process is portrayed in figure 5.

As part of the coursework, students work in teams to design the instrument, then publish their specification on a web site. They then work part-time on this project for several weeks, building their instruments. At the end, they are expected to write a critical analysis of how well the instrument works, and how it fares compared to their original specification. Finally, there is a presentation day where the whole group watches performances of the finished instruments, with or without a conventional instrument ensemble. This emphasis on a performance focuses the mind somewhat on producing a usable system that is capable of complex real-time control. In so doing the students find they have achieved what is considered to be a highly difficult and extremely challenging engineering task.

3.4. *Multiple media installations*

The MIDAS system is also used later in the course in the context of the special interest group on multiple media installations. The structure of this course unit is similar to that described above, in that groups of students design and build an

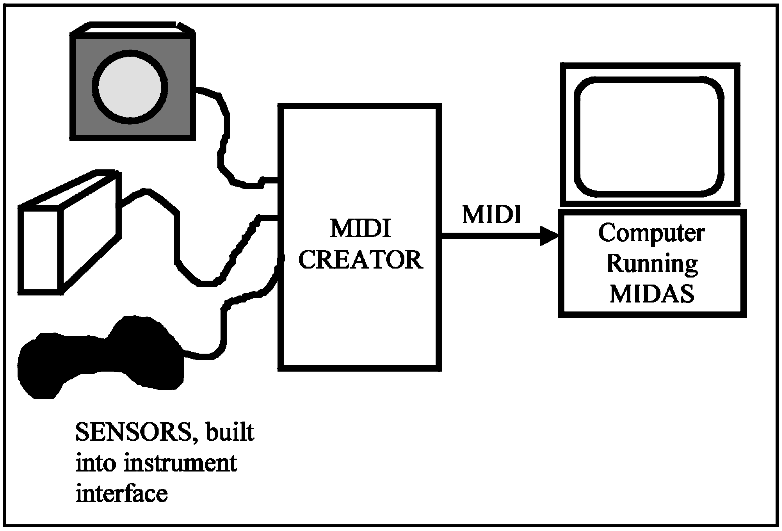

Figure 5. A new instrument, consisting of a sensor-based interface, the *MidiCreator* converter and the MIDAS synthesis engine.

interactive system. However, in this case the system is an *installation*, such as might be exhibited in a public gallery, museum, or perhaps a nightclub. The other key difference is that the system incorporates dynamically created *images* along with the sound.

MIDAS has been designed to allow the seamless integration of building-blocks from different media types. A simple example of this is now explained. Figure 6 shows a MIDAS network consisting of a Mirror UGP which draws four simultaneous triangles symmetrically around a centre-point.

This simple network can produce some quite complex images (see figure 7). The complexity comes from the mirror's ability to generate symmetrical shapes, coupled with the oscillator's slow change of image size, and the user's overall control using the computer mouse.

This simple example can be expanded so that sound and image are processed and generated, under the total (or partial) control of a human user, or even more than one user! The networks once again need to be driven from control gestures in the real world. Students' previous solutions to this problem have been to use the computer mouse, multiple channels of MIDI input via *MidiCreator*, sonic input only (talking into a microphone) and keys on the computer keyboard (disguised by covering it up with a painted cloth).

Students are encouraged to think about the space in which their installation is to be exhibited. The expectations of the passers-by need to be considered, along with a conceptual design that gives the project a coherent theme. Then come the tasks of building the interface, designing the audio-visual processing software, then linking it all together. Once again students are assessed on their engineering foresight, planning and execution as well as their artistic design skills. Recent projects have had real public viewings in the York Y2K Festival and the *LoveBytes* festival in Sheffield, UK, and more have been requested for future events.

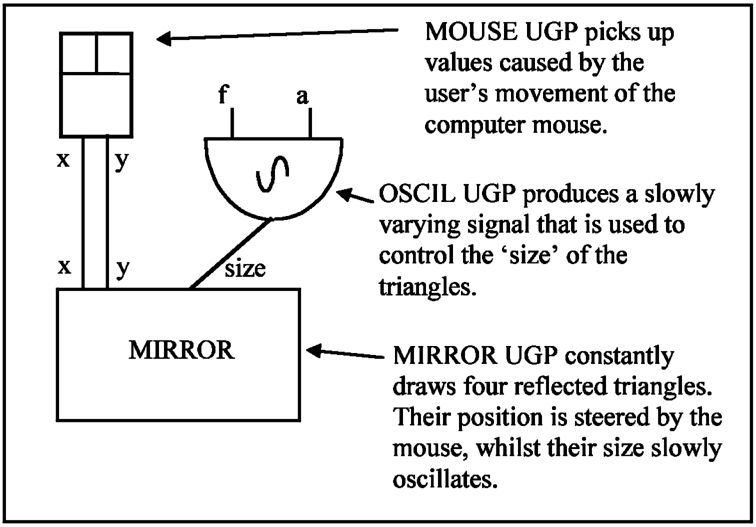

Figure 6. A three UGP network for drawing a complex shape, with user interaction.

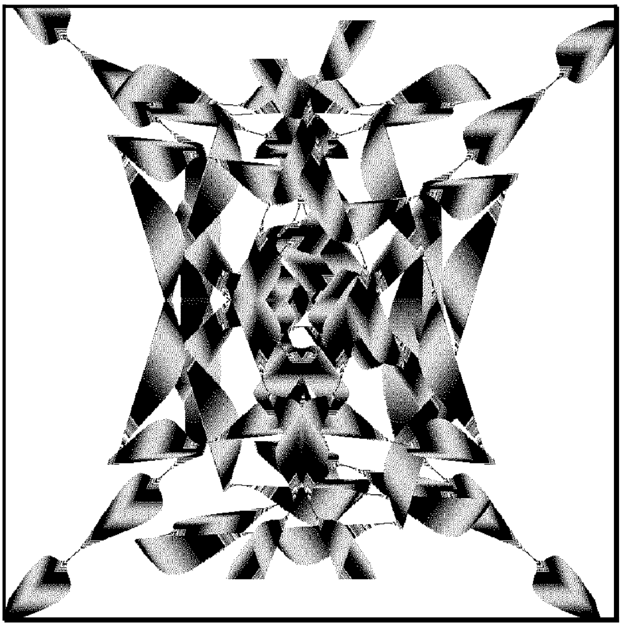

Figure 7. The image resulting from running the network in figure 6 for a few seconds.

4. Hardware systems for digital signal processing

4.1. *Introduction*

The dual goals of harnessing creativity and group work can be brought to bear in more traditional core electronic engineering exercises as applied to musical tasks. The implementation of electronic musical instruments essentially involves the

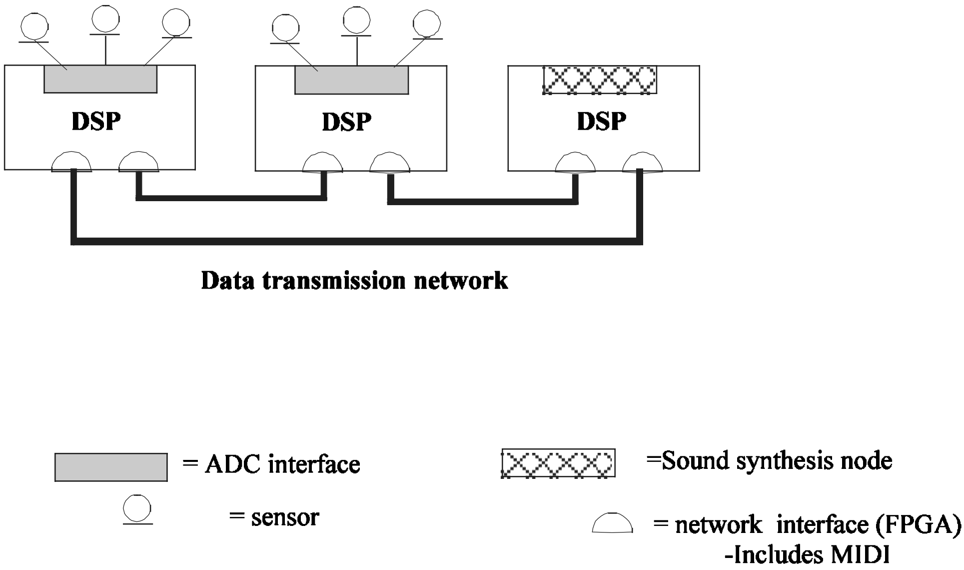

Figure 8. A ‘consort’ of electronic instruments.

application of real-time digital signal processing (DSP) techniques to synthesize or transform sound under the control of a human performer. The DSP algorithms need to be made accessible to the performer through appropriate user interfaces embedded within the bodies or shells of the instruments. The design of these new musical instruments thus provides ample scope for novel forms of integration of human–computer interface (HCI) and DSP techniques set within the context of design exercises undertaken by groups of students.

4.2. *DSP for sound generation and processing*

A typical project might involve the application of digital filtering techniques to sound synthesized by various forms of modulation, or perhaps sound sampled via a microphone, or combinations of these. The outputs from sensors forming the user interfaces might be used to control parameters such as modulation indices and filter coefficients, thus influencing the evolution of sound produced by the instrument. If a number of these instruments are to be produced, to form a ‘consort’ of electronic instruments, then network interfaces may need to be produced to form the instruments into a coherent distributed processing system. Figure 8 illustrates such a consort of instruments.

Figure 9 shows an example of a ‘shell’ instrument forming part of the consort. It has been specially constructed out of moulded fibre-glass, and is fitted with embedded piezoelectric sensors, DSPs and network interfaces.

The reason for the group approach to the exercises arises from the scope of the work to be undertaken. There may be a requirement for DSP coding for sound treatment, hardware design (e.g., for the production of network interfaces), HCI design and overall product design forming the conceptual architecture of the instrument. This amount of work and the range of specialities required are beyond the realistic capability of any one student, given the time scales involved. A group approach

Figure 9. A 'shell' instrument forming part of the consort.

provides the necessary human resources, and adds valuable dimensions regarding group dynamics and also the communication of design intention within the project.

4.3. *Hardware details of the DSP platform*

A hardware platform has been produced by the authors, which forms a flexible basis for the student's work. It consists of a Texas Instruments TMS320C31 'Starter Kit' augmented with extra memory, together with analogue and digital interfaces. A Xilinx XC4005E Field Programmable Gate Array (FPGA) is also provided, which may be used for interface address decoding and other specialized hardware functions. The unit is connected to a PC which provides access to C cross compilers, assemblers, a debugger and the CAD system for the FPGA. The CAD system includes functional and timing simulators, VHDL and schematic capture design entry, as well as device fitting and routing. The layout of the hardware platform is shown in figure 10. The front panel is shown in figure 11, and this is the student's view of the system.

4.4. *Academic issues involved in the practical exercise*

The design exercise spans two academic terms and occupies approximately 36 contact hours for each student in total. About one-third of this time is given over to demonstrations of the PC support environments (CAD and programming tools), together with set-piece DSP and FPGA design laboratories. These are carefully designed to incorporate cardinal design concepts, which will be used in the main group projects.

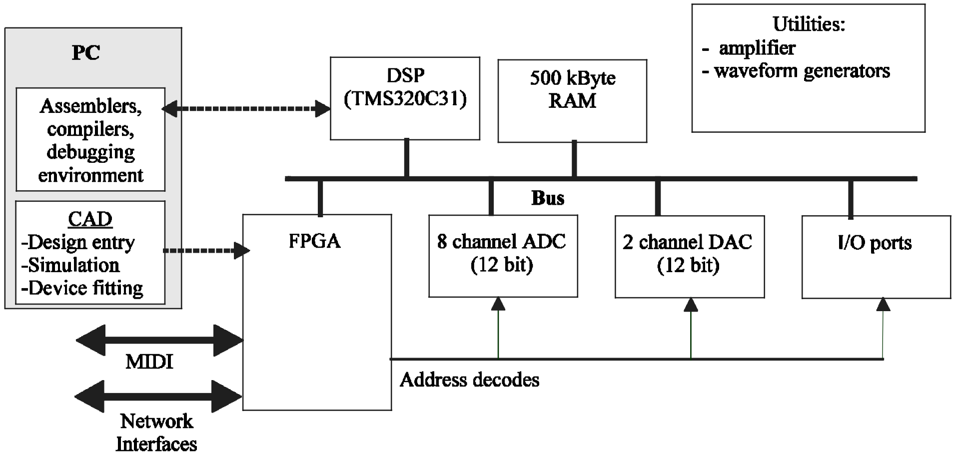

Figure 10. Hardware platform for the musical instrument design group project.

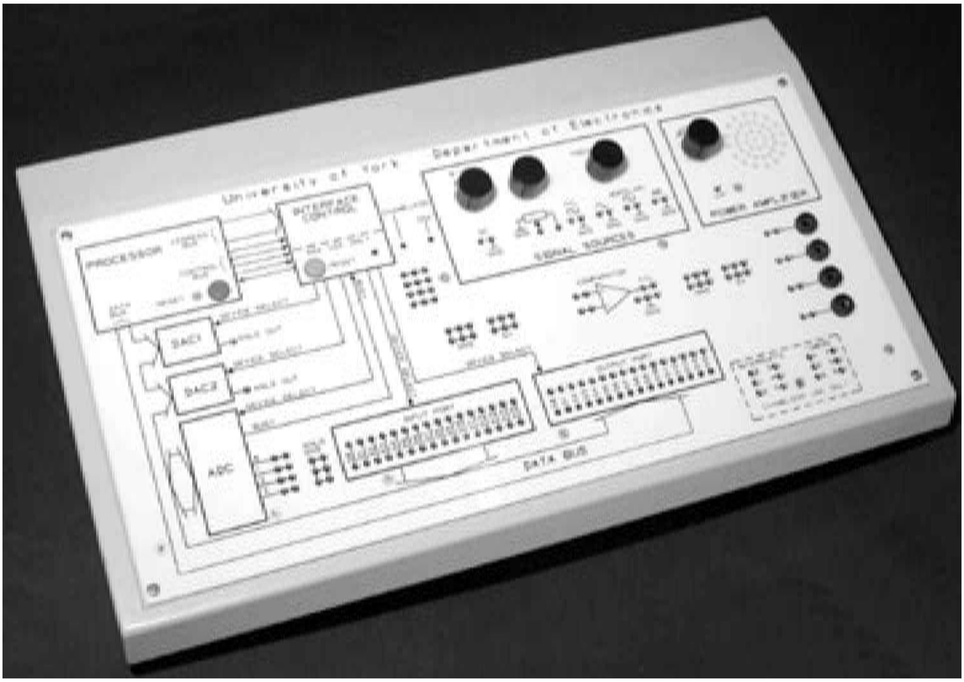

Figure 11. Front panel of the platform (the student's view of figure 10).

Students are then placed into their working groups of six to eight students and the main design task is allocated. A further 6 h is assigned for individual groups to evolve a conceptual design in response to the allocated task, which they are required to document and submit for formal review by the rest of the class. A 2-h session is given over to the reviewing process. The reviewing group has access to the relevant design documentation and receives a presentation based on the documentation by its design team. The emphasis in the review is concerned with an assessment of the likely adequacy of the design against the requirement specified by the staff running the exercise, as well as the identification of areas of technical risk. A useful spin-off is that groups see different interpretations of the design brief given to the class, and are allowed to amend their proposals in the light of their observations.

The remainder of the time allocated is used for the implementation of the designs refined by the groups, through the review process. Staff act as consultants during this time. Informal tutorials may be arranged to deal with commonly recurring technical queries. The project ends with a presentation of the completed implementation of the design by each group to the rest of the class.

Assessment is based on an element common to all members of the group concerned with the effectiveness of the conceptual design, together with individual elements of assessment based on a design log kept by each student throughout the implementation phase of the project.

Most academic courses in engineering are based on vertical strands of thematic material presented in a logical progression throughout a degree programme. Unfortunately, many students may fail to appreciate common threads that occur across these strands. Knowledge can become over-compartmentalized. To take a simple example: the operation of a transistor may be associated primarily with a strand concerned with ‘analogue electronics’, with students failing to appreciate fully the role of transistor operation in digital circuits forming part of a computer interface. A considerable pedagogical advantage arises from the approach to practical work described here because the range of knowledge required spans across traditional strands. The projects are too large in their scope for compartmentalization to take place. Students are forced to take a more holistic view of the design, recognizing that component technologies need to be integrated in synthesizing a solution to a specified problem. For many students the project thus provides a valuable opportunity to revisit concepts imperfectly understood in earlier, more conventional academic settings.

5. Software engineering project

5.1. *Introduction*

Group work also forms an important component in another kind of project associated with musical design activities, this time involved with the production of musical software. The software engineering group project brings into play issues relating to group dynamics, communication of design intention, delegation of roles and integration of individual effort within a task which is too complex and large to be taken on by any one individual. Software is chosen as the medium partly because of its relevance to the musical industry, but more specifically because of the relative ease with which it is possible to create large, unmanageable (software) artefacts. In order to control the issues of complexity inherent in the production of these artefacts within a group enterprise, students are forced to adopt a disciplined approach to design and management of the group activity.

5.2. *The project and its management*

The design activity is placed within the context of a conventional software engineering life cycle involving an orderly progression from requirements analysis, specification modelling, design by stepwise refinement, code, integration, test and maintenance. Once again, the review process is used at various points in the life cycle as a means to test the adequacy of proposed responses to the brief given, and also as a means to communicate aspects of the design amongst members of the design team.

The disciplined approach to management of the group activity is based on the development of a quality assurance (QA) scheme which lays down the organizational framework within which the group will work. One of the first activities undertaken by the group, before the software task is allocated, is the production of a QA Manual. This defines the classic QA brief: 'what is done by whom, by when, using what methods, to what standards of quality'. In synthesizing a solution to this brief, students realize that they are laying down the management framework for the management of their group. QA is thus lifted from a vague notion of bureaucracy to a practical tool for dealing with difficult group dynamics.

Students rapidly become aware that the project is not just about 'learning to program better', but involves these wider organizational issues which are widely applicable in many professional spheres.

Setting the software group project within a musical context again gives scope for motivation through creativity. Often the software task is related to a musical performance to be undertaken by the group. This provides an initial entry point for students who do not regard 'programming' as one of their principal strengths. It is not uncommon for these students to go on to take a leading role in the project, as group members realize the importance of developing the potential contribution of all members to technical and managerial functions within the group.

6. Conclusions

This paper has described three systems and several academic programmes, developed at the University of York, that allow students to engage actively with engineering principles within an artistic and creative framework. The students learn rapidly and understand the nature of the constructive design process, as well as the more common analytical approach. The students work in groups, at a non-superficial level throughout the whole design process, whilst having the option of engaging with the system designers to develop the systems further.

Engineering is a creative activity, and students need prolonged periods of directed engagement with the building blocks of their trade. We hope that this paper has stimulated some thought in this area, and has presented some approaches which may be of use to others working in this field of education.

Acknowledgements

The development of the acoustics teaching software was initiated through the support of a University of York Teaching Innovation grant and it has been further supported as part of EPSRC research grant GR/M94137, for which the authors are grateful.

References

Bragg, W. H., 1920, *The World of Sound* (London: G. Bell and Sons).

Campos, G. and Howard, D. M., 2000, On the computation time of three-dimensional digital waveguide mesh acoustic models. Proceedings of the 26th Euromicro Conference, Maastricht, vol. 2, pp. 332–339.

Howard, D. M. and Angus, J. A. S., 2000, *Acoustics and Psychoacoustics*, 2nd edn (Oxford: Focal Press).

Hunt, A. D. and Kirk, P. R., 1997, Technology and music: incompatible subjects? *British Journal Music Education*, **14**, 151–161.

Hunt, A. D. and Kirk, P. R., 1999, *Digital Sound Processing for Music and Multimedia* (Oxford: Butterworth-Heinemann).

Kirk, P. R. and Hunt, A. D., 1996, MIDAS-MILAN: an open distributed processing system for audio signal processing. *Journal of the Audio Engineering Society*, **44**, 119–129.

Murphy, D. T. and Howard, D. M., 1998, Modelling and directionality encoding the acoustics of a room. *Electronics Letters*, **34**, 864–865.

Murphy, D. and Howard, D. M., 1999a, Digital waveguide modelling of room acoustics: comparing mesh topologies. IEEE/Euromicro Proceedings of the 25th Euromicro Conference, Milan, pp. 82–89.

Murphy, D. T. and Howard, D. M., 1999b, The WaveVerb multi-channel room acoustics modelling system. Proceedings of the International Computer Music Conference, ICMC-99, pp. 472–475.

Pearson, M. and Howard, D. M., 1995, A musician's approach to physical modelling. Proceedings of the International Computer Music Conference, ICMC-95, pp. 578–580.

Pearson, M. D. and Howard, D. M., 1996, Recent developments with TAO physical modelling system. Proceedings of the International Computer Music Conference, ICMC-96, pp. 97–99.

http://www.york.ac.uk/inst/mustech/gsp/welcome.htm

http://www-users.york.ac.uk/~adh2/midas/home.html

http://mitpress.mit.edu/e-books/csound/fpage/gs/gs.html

http://www-users.york.ac.uk/~adh2/book/ch4.htm

http://www.midicreator.com/

About the authors

Andy Hunt is a lecturer in Music and Media Technology at the University of York, UK. His DPhil was concerned with radical user interfaces for real-time control, and this reflects his research interests in HCI, novel musical instrument interfaces, including those for people with disabilities. He is co-developer of the MIDAS system, described in this paper.

David M. Howard holds a personal chair in the Department of Electronics, University of York, UK. His research interests include the analysis and synthesis of speech, singing and music, and he has a particular interest in providing real-time visual displays to encourage singing pitching development in children.

Kingsley Ash is a research assistant at the University of York, working with the other authors on a physical modelling prototyping system, using tactile feedback. He graduated with distinction from the Masters Music Technology course mentioned in the paper.

Ross Kirk is a senior lecturer in music technology and microelectronics. He co-founded the Music Technology course at York in 1986. He is chair of the Ensemble Research Group, which is a forum for special needs professionals to meet with academics and industrialists. He is co-developer of the MIDAS system, described in this paper.

Andy M. Tyrrell is Head of the bio-inspired and bio-medical engineering research group at York and Head of Department. His main research interests are in the design of biologically-inspired architectures, evolvable hardware, FPGA system design, parallel systems, fault tolerant design, software for distributed systems, parallel systems for numerical problems, simulation using parallel computers and real-time systems. He has published over 120 papers in these areas, and has attracted funds in excess of £750 000. He is a Senior member of the IEEE, a Fellow of the IEE and Vice Chairman of EUROMICRO.