Sybil: Synthesis by Interactive Learning

Michael Clarke*, Ashley Watkins, Mathew Adkins, and Mark Bokowiec

Department of Music, University of Huddersfield, Queensgate, Huddersfield, England HD1 3DH
*j.m.clarke@hud.ac.uk

Abstract

This paper introduces Sybil, an interactive tool for assisting in the teaching of sound synthesis and processing. We have created a 'browser' in MSP that permits patchers to be assembled into a structured learning package. All the sophistication and flexibility of MSP is available. Sybil provides text accompanied by examples that students can manipulate in real-time, hearing the results and seeing the effect of their changes on the signal and spectrum. This link between theory and practice greatly enhances the learning experience and makes an important link from theory to the creative potential of these techniques. Sybil works on three levels: (1) there are the modules we have created, (2) being written in MSP the program is easily extensible, (3) students learning to use MSP themselves may use the patchers that generate sound in Sybil as the basis for further exploration or creative work. A demonstration will accompany the paper.

1 Introduction

Sybil was designed with our own Music Technology students in mind, although we believe it is also of much wider value to other courses and institutions.

One of the challenges we face, like many other institutions, is that our music technology students come from a variety of backgrounds. This is a positive feature and something we encourage. Some are primarily musicians with little technical knowledge, whilst others have science qualifications and are taking modules in Engineering as well as in Music. The challenge is to present the technical information in a way that is suitable for everyone, allowing those who are confident technically to move more quickly through material while at the same time giving the opportunity to those from less technical backgrounds to advance more gradually.

Another task with all students, whatever their background, is to help them relate technical information to real sounds and to the creative application of these ideas. We do not want to produce graduates who have simply studied music and science, but rather students who understand the connections between the technical and the creative.

Sybil is designed to help with both these problems. It is intended to supplement lectures and textbooks, not to completely replace them. Sybil permits students to work through material at their own pace. They can go over material more than once if they lack confidence. They can get a general overview at first and then go back over some of the more technical notes (in pop-up boxes) in more detail later (or ignore the pop-up boxes altogether if they are taking a less technical course).

Crucially, they are able to relate theory directly to sound and to visual displays of the signal and spectrum. They are able to manipulate these examples for themselves and learn through playing with the techniques. This both aids learning generally and helps to make the connection between the technical and the musical. It helps to break down some of the fear of technical issues held by some of the students from an arts background.

2 Earlier work - SYnthia

This is not our first work in this area. A previous program 'SYnthia' (Clarke and Hunter 1995) had very similar goals. However, the technology available to us in the early 1990s was far more limited than that of today (especially since we wanted the result to run on affordable multiple workstations). An earlier attempt at a new SYnthia program, somewhat similar to Sybil in concept, was limited in its success due to the hardware and software available to us at the time (Padden et al. 1996). However, that project provided a useful foundation for Sybil.

SYnthia was very successful and continued to be used in teaching at Huddersfield until last academic year. It was also used elsewhere in Britain and distributed internationally by ASK (Karlsruhe) and used as far afield as New Zealand. However, by today's standards it was limited in a number of important ways. SYnthia was written in Hypercard, using HyperMIDI to control sound generation on a MIDI synthesizer (we chose the Yamaha SY99). This both restricted usage to those who owned both Macintosh

computers and SY99s and limited the range of techniques that could be taught to those available on the SY99. The user interface provided controls and diagrams on screen but there was no possibility of displaying the signal or spectrum live. SYnthia was fixed and not easily extensible. It was more difficult with SYnthia for us to make direct connections between theoretical study using the software and the students' creative work developing from what had been learnt.

Technology has advanced greatly in the intervening years. Sybil requires no additional hardware, the sound is generated by the computer. It can also generate 'live' graphic displays, it is portable and easily extensible.

3 MSP

The choice of MSP as the language in which to code Sybil was an obvious one for us. There was no point re-inventing the wheel, and MSP provided a sophisticated and flexible aural and visual context in which we could work. It was a program we were already using as staff in our own research and creative work, and one that we were teaching to students. As a widely used program, available on different platforms, its use made for easy accessibility of our software and increased its potential for extensibility. As our students were learning to program in MSP anyway it offered the opportunity for students to see 'behind the scenes', not simply using the front end of the software but going further to find out how different processes worked.

The fact that we could make our modules into collectives or standalone applications added another advantage in availability and flexibility of use. For example, our students would be able to take copies of the software home, without the need to purchase additional software, and work on their own computers.

MSP's own extensibility is an additional advantage. Although the standard software covers most of the techniques we might want to teach, there is always the possibility, if necessary, of coding new external objects in C. The module 'FOF synthesis: an introduction' is an example of this. The external fofb~ object used in this module was coded by one of the authors in collaboration with Xavier Rodet of IRCAM (Clarke and Rodet 2003).

The choice of MSP also aids portability. Currently MSP is available for MacOS9 and OSX and for Windows XP. The original development of Sybil was done under OS9. The current modules have now been ported to OSX. This seems to have been very straightforward with few difficulties although, at the time of writing, this version is still to be fully tested. In general, however, it seems if anything to run better under OSX, some aspects of the display seeming smoother. Initial tests have been carried out with XP and again the transfer seems to work relatively easily. Further work on this is planned shortly. Currently it is not possible to build MSP standalone applications for XP, but there is no problem with making Sybil as collectives.

4 Sybil

MSP therefore provided the means necessary to generate sound and to create good user interfaces and display sound transformations visually. What was not immediately available in MSP was a means of organizing patchers and permitting users to navigate through them easily in a structured way. Our solution was to create a main 'browser' patch that then calls up the specific patchers for different topics in a bpatcher. So, in creating a module, the developer creates a sequence of patchers which are then ordered into sections, a sequence of sections forming a module. We have decided to make each module a separate standalone since they are used at different times and on different machines on our courses.

The browser then provides navigation tools which include left/right arrows for moving between adjacent pages, a menu for selecting different sections within the module, an option for going to a page directly by selecting its number and controls for the audio. All someone wishing to create additional modules has to do is to create a sequence of patchers and then use our browser to organize them. The patchers must naturally follow certain conventions but existing Sybil patchers can easily be used as templates. Clearly the size of the display has to be standardized. Audio is also routed through a central 'Audio Control' in the browser patch, so sound must be sent (send~) to the correct location. Otherwise there are no restrictions beyond what is normal in programming in MSP.

The browser requires two types of 'coll' files in order to structure the patchers into a module. The first, a single coll file entitled 'section_list', gives the number of sections in the module, the module title, and a list of the section titles in order. Secondly, a coll file is required for each section. This contains the number of pages in the section and a list, in order, of the names of the patchers for each page in the section.

It is therefore easy for an MSP programmer to adapt modules or add new modules to Sybil. Given the variety of courses and the tendency for rapid change such flexibility was a major design factor in creating Sybil.

Having created a module it can simply be used as it stands on machines that have MSP installed. We have, however, found it useful to make modules into collectives or applications. It makes it easier to transport them from

machine to machine and it makes it possible for Sybil to be used on machines without MSP installed. It means that students can't accidentally (or deliberately!) corrupt the original. We do however make many of the source patchers for the sound generation available to students. This can help them to relate the theoretical study to their study of MSP. They can explore techniques further using our patchers as a starting point. If they want to develop ideas they have come across theoretically into creative work they can again use these patchers as starting points for creative projects.

Building Sybil as a collective or application simply follows the normal procedures as described in the MaxMSP documentation. Since patchers are being loaded using bpatcher it is, however, necessary to explicitly include these in the script.

5 Two Examples

The two modules presented here demonstrate different levels of work. 'Introduction to Sound Synthesis and Processing', in the context of our courses, is intended for first year undergraduate students starting out on Music Technology (albeit with widely differing backgrounds, as already mentioned).

This module covers topics including: waveforms, envelopes, additive synthesis, amplitude and ring modulation, basic frequency modulation, filtering and an introduction to convolution.

Some of the students will have had very little previous experience of the theory behind music technology. So this module tries not to make any assumptions. It also tries to make the theoretical seem relevant and attractive to those whose primary interest is musical. At the same time more detailed information is sometimes given, often in pop-up boxes, for those for whom it is appropriate. Students who require more detailed technical information can use Sybil alongside more technical lectures or be referred to textbooks.

At this elementary stage the linking of theoretical explanation with sound and live graphic representation of the signal and spectrum is particularly helpful. For example, in learning about amplitude modulation students can not only read about sidebands but also see them, and see the effect of adjusting the carrier or modulation frequencies on the signal and spectrum. In learning about FM, students can both hear and see the effect of using an envelope to change the modulation index of a sound as it evolves. The more complex later examples begin to produce more interesting sounds that students might want to use as a starting point for further creative exploration.

The section on filtering and convolution is perhaps less technical in orientation. It provides an opportunity for students to begin to discover the rich potential of transforming recorded sounds and cross-synthesizing them. These patchers load external soundfiles, recordings of a variety of sources from a cello to a paper bag!

'FOF synthesis: an introduction' deals with a rather more complex and specialist technique. Here the problem in the past has been getting students to understand and remember a complex technique with many parameters. The interactive, self-paced approach is equally valuable in this situation. This module is used to support teaching towards the end of the second year of our undergraduate course.

Sections in this module comprise an introduction to the parameters of FOF synthesis, a study of how FOF synthesis can be used to construct a vocal imitation, and an exploration of the granular aspects of synthesis. Again it is particularly useful for students to be able to see the spectrum change as they adjust the parameters of formant regions. Later they can see how these same parameters affect the shape of individual impulses in a granular context. At the end of the section on building a voice on-screen controls allow them to experiment with different vowels, and varying amounts of vibrato and jitter. In doing so they not only learn about sound synthesis but also discover some of the factors that go into making a sound lively and interesting.

6 Evaluation

Sybil been tested thoroughly with our own students. A group of 20 first year students have worked with the module 'Introduction to Sound Synthesis and Processing'. This included testing of a prototype and further trials of the final version. Feedback took the form of questionnaires filled in by students after they had worked with the program and small group discussions. Responses were very positive to the general concept and realization of the project. Even where Sybil repeated material some of them had learnt previously, many said they learnt new things, or gained a new perspective on it from using Sybil. Negative comments about the original prototype (of which there were few) concerned issues of general appearance (we have used more colours and brightened it up in response!) and the font size for the main text (which we have increased). In general, Sybil seemed to be appreciated not only by those with a less technical background but also by those who had studied sciences who often seemed to value the links it offered between theory and real sound, and therefore creative potential. The module 'FOF synthesis: an introduction' is currently being tested by second year students. Again initial responses have been positive. Overall we were pleasantly surprised by the strength of positive feedback and grateful for a number of helpful suggestions.

7 Future plans

Sybil will be introduced fully into teaching at Huddersfield in the academic year 2004-5. It will be used in a variety of modules where the link between creative work and theoretical understanding is important. Sybil is not intended to be complete and fixed. The whole point is that it is easily adaptable and extensible. Our intention is to build on what has already been produced. The Sybil browser can be used to organize any group of patchers, so quite different types of material and approaches can easily be incorporated. We hope a number of staff at Huddersfield will contribute material each in their own style. As different modules in Sybil are separate this is not a problem. Indeed we see this as an advantage, providing variety, and meaning that different topics can be taught as is most appropriate.

We do not want to restrict Sybil to Huddersfield however. We hope the material will be widely used and we hope others will use the Sybil browser to create their own modules. We envisage its use could well go beyond Music Technology and that other areas such as acoustics and music psychology might benefit from this approach. We hope a library of Sybil modules will be shared internationally.

8 Conclusions

Sybil offers a means of arranging MSP patchers and navigating them to produce structured courses. It can help students make musical sense out of theoretical study. It is flexible and easily extensible. Its basis in MSP makes it portable and provides a wide range of synthesis and processing techniques (itself extensible) together with good and varied user-interface options. This is an open-ended project in which we hope many others will become involved.

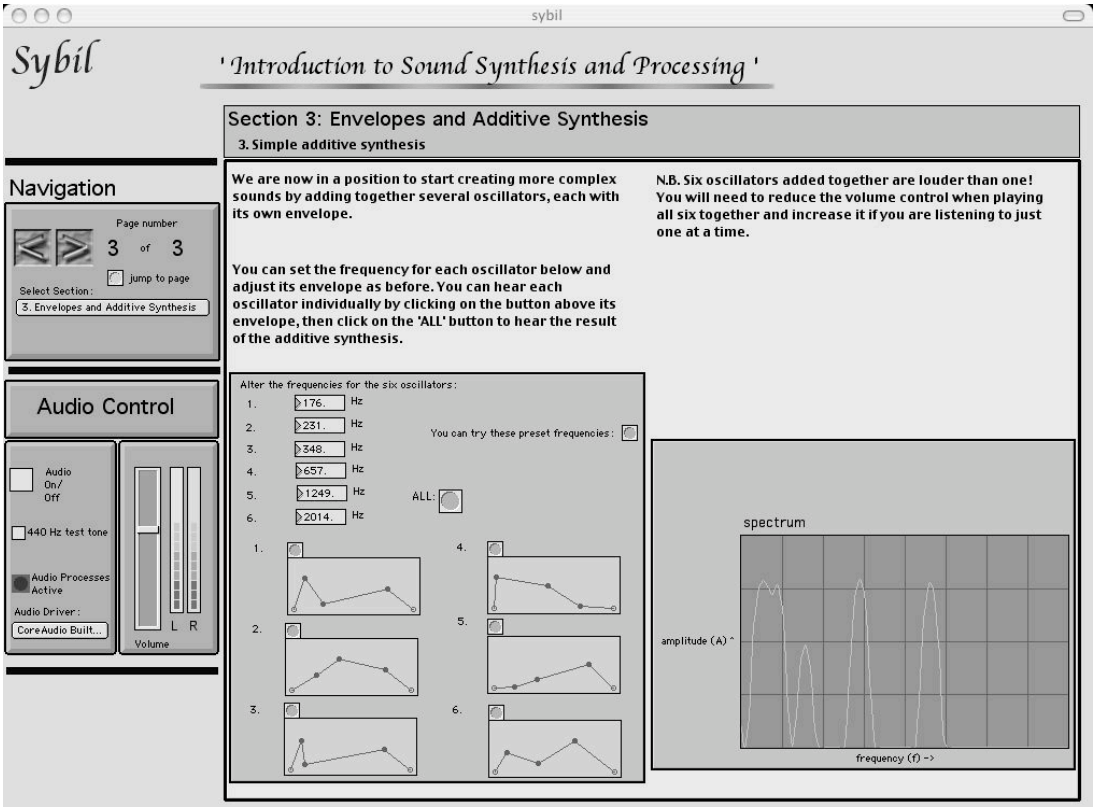

Figure 1. A page from Sybil.

References

Clarke, J. M., and S. Hunter (1995). "Educating the next generation: integrating technological skills with artistic creativity in computer music courses in Higher Education." *Musicus,* pp. 47–52.

Clarke, J. M., and X. Rodet (2003). "Real-time FOF and FOG synthesis in MSP and its integration with Psola." In *Proceedings of the International Computer Music Conference,* pp. 287–290. San Francisco: International Computer Music Association.

Padden, M.S., J.M. Clarke, A.J. Dix and M.A.R. Kirby. (1996). "Towards SYnthia II: an assessment of design strategies for Computer Assisted Learning of sound synthesis." In *Proceedings of the International Computer Music Conference,* pp. 214–215. San Francisco: International Computer Music Association.