The POOman project

Nicoletta Fornarini
Conservatorio di Padova
<nich@nextt.unipd.it>

Graziano Tiato
Centro di Calcolo d'Atearoa
Università di Padova
<tz@nextt.unipd.it>

Michele Manfrin
Centro di Sonologia Computazionale
Università di Padova
<mi@dei.unipd.it>

Abstract

The aim of the POOman project is to constitute a full-fledged public domain library of software objects written in the C++ language devoted to musical applications.

Although this library is infinitely extensible using the well-known object-oriented derivation paradigm (Cox), this paper introduces the use of this library presenting a basic set of multi-threading object tools, a small set of signal processing objects and a small set of midi objects.

1. Introduction

The purpose of the POOman project is to build a library of software objects to handle audio and MIDI in modular and natively machine-independent ways.

POOman stands for Programmer's Object Oriented Library for Musical Applications. As the name indicates, it is not a full-blown application; rather, it is a library which is supposed to be used to ease a number of recurring problems encountered designing music and audio applications. As far as audio is concerned, any application based on the POOman library should also be easily ported between different systems and be able to handle all major audio and data formats. The main attention behind the basic design of the POOman library is its extensibility and its portability; in this way, we hope that the POOman project will be able to expand in the future handling new formats, hardwares and operating systems.

The POOman library is completely written in C++ (Stroustrup; Ellis et al.). Encapsulation and information hiding is widely used throughout the library in order to isolate system-dependent code from the user interface.

2. Structure of the POOman library

Currently, the POOman library is structured as follows:

```
    PTOOLS
   /    \
P_MIDI  P_SIGNAL (ecc.)
   \    /
     (ecc.) (ecc.)
```

Figure 1. The POOman library structure
There are three basic sections of the library whose functions are:

P_Tools this is the the toolbox of the library, providing many basic building blocks used by the other sections: in particular, the P_Tools portion of the library provides a lightweight implementation of threads which uses any underlying thread implementation that the system provides: if the operating system does not provide thread support, a portable interface to the machine-independent LWP implementation of threads (Cran8, 1994) is made available in the PTools portion;

P_Signal provides all objects dealing with audio signals, including:
- audio file readers and writers
- A/D/A handlers
- audio generators
- filters
- etc.

P_MIDI provides objects dealing with MIDI messages, including:
- incoming and outgoing MIDI handlers
- MIDI file handlers
- MIDI message objects
- etc.

3. Future Developments

The Poolman library will soon be made available in source in the public domain1. Extensions, portings and applications using the Poolman library will be encouraged.

4. Acknowledgements

Many of the designs included in the Poolman library come from discussions with a number of engineers and musicians. We would like to thank here all of those who help was fundamental: Ivanos Boiscolo, Roberto Cavazzana, Giovanni DePilli, Mauro Graziani, Andrea Pravaglia and Alvise Vidolin.

5. References

Cox, B. Object-Oriented Programming, Addison-Wesley, 1987
Cran8, S. The REX Lightweight process library, 1994 available from jsc@doc.ie.ac.uk

1. Currently, the beta distribution can be requested via e-mail to nclb@net.it.unipi.it.