TEMA: An "Open" Support Environment for Music Composition

Barry Eaglestone
University of Bradford
Bradford, UK
B.Eaglestone@comp.brad.ac.uk

Peter Lundén
Royal Institute of Technology
Stockholm, Sweden
ludde@kior.kth.se

Abstract

TEMA is a generic environment for composition. It has been designed to host collections of software tools and environments and to provide generic services to assisting the composers with the management of resources, products and processes. This paper overviews the main features of the architecture and data model.

1 INTRODUCTION

The experimental and individualistic nature of computer music composition has inevitably resulted in a proliferation of software tools and environments. A consequence is that it is often difficult to utilise diverse combinations of composition tools or to keep track of the objects, tools and processes being used during composition. This situation is mirrored in other "design" disciplines, such as CAD and Software Engineering, and we have looked to those disciplines for solutions. Taking this approach, we have adapted engineering design systems technology such that it can assist artists. The resulting prototype system, called TEMA, is an "open" support environment for computer music composition.

TEMA has been designed to host arbitrary collections of composition tools in such a way that they can be used in conjunction and to provide generic tools and services to support the management of design objects, tools and the processes by which they are utilized. These services are based on support requirements for computer-based design activities, which are well established in other disciplines such as engineering and architecture. However, we have adapted these requirement to take into account the individualistic, intuitive and experimental nature of composition. In general, TEMA has been designed is such a way that it avoids imposing or prescribing the ways in which composers must work, but instead assists them with managing the by-products of working in the way in which they wish. This paper overviews the architecture and object model of TEMA. A description of mechanisms for integrating diverse sets of composition tools and environments can be found in [EL96].

2 ARCHITECTURE

The architecture of TEMA is a conventional design system architecture (see Figure 1), apart from the addition of signal processing platforms for the production of sounds. A public tools interface (PTI) acts as the integration agent for tools and data objects which are represented in an object-oriented (OO) database that acts as the repository.

An advantage of using database technology is its support for shared concurrent access to the data objects, and data independence, whereby the logical representation of data is independent of the ways in which it is physically stored. We chose to exploit OO database technology, after frustration with an earlier prototype based upon an extended relational system [EDU92]. Particular advantages of using an OO system are that inheritance and late binding allow homogeneous access to similar design objects (e.g., sounds), even though they are represented in different ways (e.g., additive, fn, granular, sampled, ...).

Tools hosted by TEMA work together at different levels. Data integration of tools is through sharable abstractions in the repository. Control integration is through the PTI which allows tools to interact as objects within the object model. In addition, a degree of process integration is achieved through constraints imposed by workspace configurations [EDU92] and by integration of tools with the data objects upon which they operate [EL96].

3 THE OBJECT MODEL

The repository is implemented as an OO database, and includes generic classes for representing organisational collections of design objects and tools. OO databases capture both

ICMC Proceedings 1996 419

Eaglestone & Lundén
structural and behavioural semantics of the environments within which they operate, and this has been exploited in order to integrate into the repository itself both generic and application specific operations upon data objects (see EL96). In addition, generic tools for managing the environment within which the computer works, and also composition tools, can be integrated within the system via the PTI. The PTI acts as an integration agent through which programs can access and modify design objects stored in the object database and also communicate with other programs, and thus allows tools interoperability.

![Diagram of TEMA architecture](image)

**Figure 1: Outline TEMA architecture**

Design objects are the artifacts which participate in a composition activity and are organised within hierarchically structured workspaces, i.e., named repositories of design objects, within the environment's repository. The generic services are concerned with management of design objects, but these can be specialised to represent application-specific artifacts, such as sounds and dance. Workspace can also be specialised to support different methods of composition. For example, in the original prototype workspaces were specialised as archivist, palettes and scores to reflect one of the composition methods used by Tamás Ungvary [EU92]. As extended version model [EDR93] allows different versions of design objects, generated during composition, to be retained within the system, together with annotation, for example detailing how they were created. This facility supports experimentation with both design objects and composition tools. Versioned design objects are represented within event graphs [EHR96], whereby the edges link versions to other versions derived from them, and edges are tagged with representations of the events which caused the generation of the new version. Where possible, details of the generative events are captured automatically and can be re-applied in other contexts. The event classes enable design objects to carry with them their history and documentation.

5 IMPLEMENTATION NOTES

TEMA has been implemented to run in a UNIX/Motif environment. The implementation uses two public domain software systems: OBST is an OODB management system, and is used to implement the repository; ILU (Inter-Language Unification) is used to provide the PTI, and thus allows composition tools written in different programming languages to be integrated. At present we are able to host tools written in Lisp, C and C++.

6 CONCLUSIONS

The approach that we have adopted seems to us an obvious one, since it brings to the computer music community benefits already provided for other "design" disciplines. The OODB technology that we have chosen to use is extensible, and so we anticipate a natural evolution and expansion of the capabilities of such an environment as it becomes established and the capabilities of tools are "absorbed" into the system. We also plan to widen the scope of the generic services provided. Specifically, we are currently researching support for cooperative composition activities, and for extending the human interface capabilities.

REFERENCES


