

# IA-SIG Meeting Summary

*Date: February 21, 1997*

Summary by Mark Miller and Brad Fuller  
Minutes taken by James Grunke



# IA-SIG Meeting Summary - February 21, 1997

## Attendees

James Grunke	Onlive technologies	james@onlive.com	408 777 2146
Chris Van Rensburg	Headspace	chris@headspce.com	415 696 9415
Thomas Dolby	Headspace	thomas@headspace	415 696 9410
Brad Fuller	Matter to Magic	bfuller@m2magic.com	408 377 9372
Jim Reekes	Apple	reekes@apple.com	408 974 5374
Jeff Roberts	Rad Gametools	jeffr@radgametools.com	801 322 4300
Rudy Helm	At the Helm	rudyhelm@sierranet	209 683 3704
Albhy Galuten	Universal Music Group	albhy@unistudios.com	818 777 8963
Fred Mack	Consultant	fred@metalithic.com	510 651 5123
Tom Hays	Rocket Science	tomh@rocketsci.com	415 536 5756
Alex Stahl	Pixar	alex@pixar.com	510 412 6008
Heather Gold	Apple	gold@apple.com	408 974 2954
Jonelle Adkisson	Broderbund	jonelle@broder.com	415 382 3059
Daniel Bernstein	Monolith Productions	dbernstein@lith.com	206 889 4532
Donald Griffin	Computer Music Consulting	donald@computer-music.com	415 285 3852
Jim Hedges	Consultant	mcigor@ix.netcom.com	415 282 4034
Rob Hubbard	Electronic Arts	rhubbard@ea.com	415 513 7551
Mark Miller	Crystal Dynamics	mmiller@crystald.com	415 473 4929
Tom Miley	Sega	tom.miley@segasoft.com	415 802 6213
Eran Egozy	Harmonix Music Systems	eran@harmonixmusic.com	617 491 6144
Gary Levenberg	Interactive Audio	gary@interactiveaudio.com	415 512 5510
Michael Land	Lucas Arts	mland@lucasarts.com	415 444 8388
Todor Fay	Microsoft	todorfay@microsoft.com	206 703 1454

## *Why have this meeting?*

We met because, as a group, we think that there is value in exploring a common approach to the creation of Adaptive Audio Systems and the possibility of creating standards where Adaptive Audio Systems could exist.

## *What is Adaptive Audio?*

We agreed upon the following definition:

***Adaptive Audio:*** *Audio that is delivered via a system that allows for direct or indirect control of the data and/or data stream.*

(In this context 'data stream' refers to the fact that the data is being played back at the time of the manipulation or control in real time)

## *Why is this important?*

It is important because we have a vested interest in furthering the state of the art in audio for interactive media. It was stated that the main goal is to provide the consumer with a sonic experience that rivals and even excels the impact of an excellent film score.

We discussed that the following was also important:

- Create a large enough market of content providers who will utilize this technology so that the tool smiths see an opportunity and create powerful and easy to use tools for creating Adaptive Audio Content.
- Create an architecture that will allow for different systems and playback engines to work together within the same product. This

will open up the creative possibilities for content providers and lower the work load on application programmers.

- Make it easy for developers to adopt and integrate this new technology.

## *What is the value of an Adaptive Audio Systems and to whom?*

We discussed that a simple direct sell statement targeted at both consumers and producers was needed. Thomas Dolby suggested the following:

“ (Adaptive audio systems provide) a heightened user experience through a dynamic audio soundtrack which adapts to a variety of emotional and dramatic states resulting, perhaps, from choices the user

makes.”

## *What form should our results take?*

This question came up rather early on in the meeting and was answered through the natural course of the discussions. We ended up with ideas for an API and a ‘Plug in’ style architectural map for implementing Adaptive Audio Systems. Also, an ‘object oriented’ approach was discussed.

## *What are the basic challenges in creating an Adaptive Audio System?*

We discussed how to deal with time. In our meeting. This issue was addressed on two levels. The first level is how time relates to nonlinear media and composition. The

second, addresses the need for a common time base (i.e. a Master Clock) for different Adaptive Audio technologies and paradigms. Following is a brief summary of those two discussions.

1) Since Adaptive Audio systems involve modifying data to produce non-linear music, additional tools for production are required (in addition to the traditional tools of composition.) As the media becomes non-linear, new elements and structures must be added to the 'composition' so that it can adapt and navigate the changing landscape. These new paradigms imply that time becomes non linear and unpredictable. (for instance, you can't just 'spot' the event you want to build up to and then construct a clever musical passage to achieve your emotional goals.) Thus, Adaptive Audio Systems must integrate both musical data and control functions so that the emerg-

ing flow of events can be 'scored' on the fly and keep the internal time integrity of the music in tact.

2) All concurrent Adaptive Audio players (i.e. players that must play their data at the same time, synchronously) must read the same master clock. This function was later delegated to the component of our prototypical system called the 'Adaptive Audio Engine'. It was also discussed that it would be beneficial to employ a common time base (such as 480ppq or 44100 sample per second) that all systems would use as their basic unit of time. Thus noted, this idea was put aside for later consideration.

*So, how do we begin to talk about these issues?*

What followed was a somewhat disjointed discussion of terminology. The meeting almost ground to a halt because of the philosophical issues defining terminology before functionality is described. There was fear that the terms defined at the outset would limit the discussion of functionality into the intellectual framework implied by those terms. Most agreed that the definition of common terms should come first so that one could discuss functionality in any civilized way. In the end we decided that the final terminology used to describe Adaptive Audio System functionality will be determined by history, not by a meeting at this time.

## **Terms/Functionality**

We decided that an 'object oriented' model made the most sense. It was decided that there would be two basic types of objects: "Control Objects" and "Data Objects."

Control Objects were more briefly touched upon, and generally referred to as objects that contained scripts or traditional code. Their function would be to control the behavior of Data Objects.

Within the class of Data Objects there would be some subclasses. One subclass would be the 'Segment Object'. The Segment Object would be defined as "a bounded chunk of time based data". Time base was important here because a segment was to refer to more traditional music with some internal time reference or time stamps. However, it would

go beyond music data to include commands, callbacks and opcodes (as long as these new types of data were referenced to time in some way.) It was also discussed that there should be a Data Object whose function was to contain a collection of other Data Objects. A “Track” was defined as a “horizontal subdivision of a Segment” like a MIDI channel.

## Review of IAE 2.0:

Next we moved on to the functionality section of the IAE 2.0 document.

Here are the main points:

### ***General functionality:***

1: “Multiple Segments...” was broadened and revised to read “Multiple Data Objects must be able to be played back simultaneously and in synchronization if so desired.”

Definitions for ‘callbacks’ and ‘registers’ (items 2 and 3) were skipped over.

4: “Different data types (digital audio, MIDI, algorithmic generator data, etc.) will have specific, but not necessarily identical, parameters associated with each Segment.

# IA-SIG Meeting Summary - February 21, 1997

These parameters will be stored and updated as the Segment is played and can be changed in real time.” The term “Segment” was revised to “Data Object”.

This led to a discussion about how each Data Object would ‘publish’ a set of parameters to the rest of the system. Changing these parameters would be the method for controlling a Data Object’s behavior in real time.

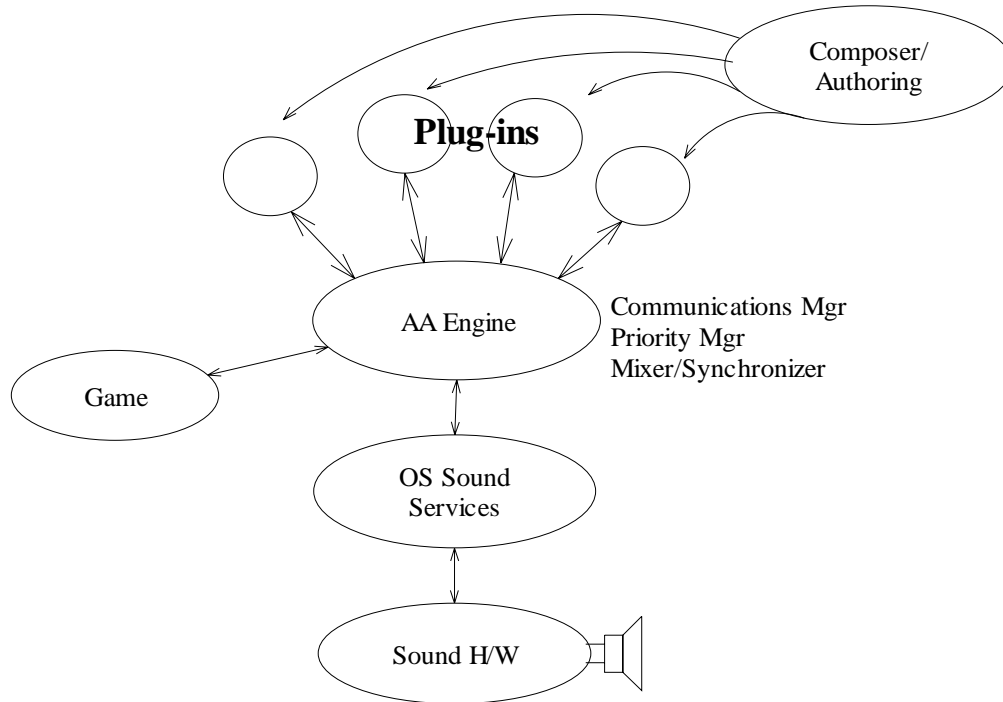
In the section on Specific Functions and Flow control set, it was decided to change “play segment” to “start object”. “unpause” was changed to “resume”.

“Install Modify Hook” was removed from the list of parameters and added to the list of general functions under the ‘modification set’ along with “Set Parameter”, “Get Parameter”,

“Change Parameter over Time”, “Send Direct MIDI command”. The “Install Modify Hook” was defined broadly as a function that would allow for the calling of a custom function from within a Data Object or Control Object or directly through the API to operate upon either data or the contents of a Control Object.

It was here that the ‘plug in’ idea started to take hold. The diagram on the next page was placed on the board as how this might work. In this model, the composer creates data with an authoring tool specific to the AA driver that they plan to use. The AA driver comes in the form of a plug in that feeds its data to the central Adaptive Audio Engine (AAE.) The AAE may take care of priority, timing and communication issues. It was suggested that different plug-ins could be made to work together. Plug ins might also be

# Plug-in Architecture Suggestion



## Issues:

- Shared memory resources between plugins
- Requires Traffic Cop
- Get Parameters - which one, from where?

data processors, in which case the output of one plug in could be piped into the input of another plug in to be processed and passed on to the AAE. The AAE would finally mix and output the data to lower level sound services such as Direct Sound. Different systems would be easy to substitute throughout a project without requiring massive re-coding efforts.

Items in need of greater definition and discussion are as follows:

- The Plug in format
- Adaptive Audio Engine (AAE)
  - Its role as a communication manager, a priority manager, a mixer/synchronizer
  - Its Interface to the Game and to plug ins
- Clock and timing format, including time stamping and latency offsetting. What should be used as the Maser Clock? Vblank, hblank, DAC?
- Command format and protocol
  - Get, set change over time, open object, close object, get params, enumerate, params, poll, Set params.
- Communication protocol
  - Plug in registry containing query-able parameters (the items above)
- Need for emulation tools for authoring that can

# IA-SIG Meeting Summary - February 21, 1997

simulate run time environment

Finally, the following questions were raised:

Who will do this work? It was suggested that there were enough people in the room that needed this to be done and would support the effort on a volunteer basis. If this failed, other options would be looked into. If we needed to hire some one to do the work, who would fund it and who would own it? It was decided that the IASIG could coordinate the funding from member of the ICWG and would then hire the contractor and own the results. It would be set as a condition that the IASIG would declare from the start that all information developed in this way would be made public domain.

If the ICWG did not hire the contractor(s), who would own the resultant work?

It was lastly decided that this group would meet again in 6 weeks to take the next steps.

