Current state of SpatDIF
"Vacuum Cleaner Bag Phenomenon"

... is it useful to have self-contained, non-exchangeable formats?
SpatDIF

Spatialisation: for spatial audio
   music? acoustics? audio scenes?

Description: meta descriptors
describing what?

Interchange: interoperability
software agnostic

Format: what format?
   OSC, XML, YAML, SDIF, JSON, ...?
SpatDIF so far

• Peters et al. (2007):
  • Reviewing available spat apps
  • Self-contained, non-exchangeable description and formats
  • Need for standardization for interchangeability

• Previous meetings
  • Workshop @ BEK, Bergen 2007
  • ICMC Panel Discussion, Belfast 2008
  • Meeting at CIRMMT, Montreal 2008
  • Workshop @ GMEA, Albi 2009
SpatDIF is not

- A general music notation system system (such as MusicXML)
- A sound synthesis language
- A 3D graphic format
- Primarily made for computer games (such as OpenAL or irrKlang)
Use Cases

When developing SpatDIF, we have the following use cases in mind, they help to guide decisions and clarify the scope of what is to be described.
Karl-Heinz, the Composer

• Creates spatial, electro-acoustic compositions using DAW
• Wants to perform pieces in different venues, adapt on-site
• Inadequate tools for compositional process
  • Limited way to describe and create space
  • Limited number of loudspeaker channels
• Needs a format to store and conserve his ideas
• Pieces performed in different halls, with different equipment, computer hardware and loudspeaker configuration. Wants pieces to be reproduced as ideally as possible.
Pierre, the Sound Installation Artist

- Works mostly on sound installation in gallery spaces and acoustically unusual environments
- Uses different spatial sound renderer in different software environments
- Often uses irregular loudspeaker setup, far away from ITU recommendations
- Creates spatial sound scenes, tries out how different sound renderer perform on site
- Every installation is unique, and he wants to document it.
- Installations are often interactive
Delia, the Sound Engineer

- Produce musical content in consumer formats
- Works mostly with a DAW (Digital Audio Workstation), lot of plug-ins
- Uses DAW internal automation features for spatial scene manipulation - time-consuming
- Transfer of projects to different DAW is complicated as automation data are usually lost if a 3rd party plug-in is not present at the other DAW
- Every 5 years she has to re-arrange her projects for yet another better format
Francis, the Acoustician

- Studies human auditory perception in audio reproduction system
- Wants to perceptually evaluate spatial sound renderer and multichannel audio codecs
- Researches methods to synthesize and manipulate perceptual attributes, such as source width or listening envelopment
Pauline, the Virtual Reality Researcher

- Uses the internet as a performance space
- Organizes telepresence concerts with remote musicians
- Often musicians and audience are placed freely in a virtual environment and auralization techniques are used to simulate a specific room's acoustic
Leela, the Musicologist, year 2090

- Likes spatial music from the beginning of the Millennium
- Often can't find a reproduction system to play back historical tape music, such as DVDs or DAW-projects files with loudspeaker associated audio files
- Wishes that composers had used a scene description notation independently from the reproduction setup, a scene description she could study now more easily
Edgar, the avant-garde visionary composer

Edgar doesn't like bullets in text writing and finds current contemporary spatialized music to be naive. He thinks that what most composers do with sound in space resembles what a child would do with a pen and pencil: Draw simple geometric figures like dots, lines, curves and spirals. With current tools this is all that is possible, but Edgar dreams of new ways to dream about spatial sound, new ways to notate, analyze, edit and manipulate spatial information that would open for new understandings and conceptions of sound in space, in a similar way to how the introduction of notated music formed a foundation for the development of western musical style in the last millenium.
Olga, the spatial sound sculptor

- Wants to extend periphony to pluriphony, i.e. work with a multiplicity of simultaneous sound sources from multiple directions
- Compose with spatial scattering and spectral diffusion techniques
- Looking for ways to manage and describe the large amounts of control-data required.
- With current tools this is difficult to achieve in terms of logistics, modellisation and control-interfaces.
Who did we forget?

• Ennio, the film score composer?
• Nolan, the game sound designer?
• ...
Requirements

- Easy connectivity with editors, interfaces and controllers to create spatialization in multiple ways.
- Multiple layers of interaction to control and explore spatial features from different higher level viewpoints.
- Human-readable syntax to prevent misunderstandings when exchanging stored data.
- Real-time control of spatialization is desired to explore the possibilities and interactions within the virtual space through receiving immediate audible feedback.
- Non-real-time applications supported, such as OpenMusic.
Requirements II

- Free and open source to increase the acceptance and widespread usage of the new format.
- Extensibility is important as long as the format is under development, but moreover, to adapt to new developments in audio technology and compositional styles.
- Platform independence permits audio scenes to be exchanged. Any 3D audio rendering algorithm on any computer platform should technically be able to interpret this format.
- Artistic flexibility is paramount to allow creative diversity. Limitations would cause users to reject it.
Structure of SpatDIF

Three different domains can be distinguished, where SpatDIF needs to be defined:

• Semantic
• Syntactic
• Implementation
Semantic descriptions

WHAT needs to be described

- levels of abstraction
- relationships between entities
- entities themselves
- time / space
Syntactic descriptions

HOW it is described

• namespace structure
• descriptor scope
• uniqueness of descriptors / polyvalence / overloading
Implementation Considerations

What needs to be implemented and how
• in authoring tools?
• in rendering tools?

technical issues
• issues of resolution
• time-sampling/bandwidth
• synchronicity
• interpolation
• cueing etc.
Spatialisation Workflow

1. Physical Devices
2. Hardware Abstraction
3. Decoding
4. Encoding
5. Scene Description
6. Authoring

Communication Streams:

A. Audio Data
B. Decoded Audio Stream
C. Encoded Audio Stream
D. Render Instructions
E. Scene Control Data

Processing Layers
SpatDIF schema 1
SpatDIF Basic Data Flow
with extended private Functions

Creator/Controller

SpatDIF Writer

SpatDIF stream or File
Descriptor types:
Core
Extended
Private

read SpatDIF stream to
recreate session and
extract non-renderer
specific meta/ data

SpatDIF

re-write file with
renderer specific
information/annotations
/meta-information

SpatDIF Parser

Renderer

SpatDIF schema II
Namespaces

Core and Extensions

for SpatDIF to be interoperable with spatial audio renderers of unknown capability, the implementation of a minimum set of requirements has to demanded.

Use cases:
- "the one-armed bandit" - can play audiofiles but unfortunately only in mono....
- "SupaRenderer" - knows everything about room-acoustics, virtual and real sound sources and x-amount of speakers.
Namespaces

Core descriptors are mandated.

Extension descriptors are optional.

Private extensions are just that: private

Different Applications have different needs.

(and might want to store additional data in a SpatDIF-file for convenience reasons)
Core Entities

The core descriptors can be assigned to different entities. The following entities are proposed:

- **Source**, a virtual sound source, emitting sound into the scene.
- **Listener**, a virtual sound sink, receiving sound from the scene.
- **Loudspeaker**, a real sound source, outputting sound from the scene into the real world

The index of these entities starts with the number 1
draft SpatDIF OSC-commands would be:

```
/spatdif/core/source/4/gain -12.0 db
/spatdif/core/listener/1/position 0.5 -0.5 0.0
/spatdif/core/speaker/8/position -67.5 0.0 1.0 aed
```
Core Descriptors

Position
- a poll was held to determine what coordinate system to use

Donnerstag, 20. Mai 2010
Core Descriptors

Gain
- a poll was held to determine what *gain units* to use

![Poll: SpatDif Gain Units](chart.png)
Core Descriptors

Distance Attenuation

(not defined yet)
A distance according to the position data should be simulated by an attenuation according to the inverse square law alternatively, other distance functions can be applied, e.g. to adapt for different listening environments it is not decided what these alternative distance functions are.
Extensions

Extensions are not part of the core functionalities, so that *renderers are not mandated* to understand this information. They can be considered as proposals.

- Media Extension
- Event Extension
- Ambisonics Extension
- Acoustic Spaces Extension
- Directivity Extension
- Geo-transform Extension
- Time Extension
- Binaural Extension

The most commonly used ones might become included into the core in future SpatDIF versions (via a consensus process of the community)
Extensions

Example: Media Extension
The function of this extension is to define not only where sources are spatialised, but also to assign content (media files, live inputs, internet streams) to a virtual sound source position.

Descriptors:
draft SpatDIF OSC-commands would be:

/spatdif/media/source/1/type adc
/spatdif/media/source/1/channel 1

/spatdif/media/source/2/type file
/spatdif/media/source/2/path /path/to/my/audiofile.wav
Extensions

*Example:* Time Extension

Describe a standard set of time transforms:

deals with:

- timestamps
- modifying the timebase
- setup time-based processes

Frame-based stream storage in files needs a way to describe time. Timestamps should be used on blocks of simultaneous spatDIF information.

/spatdif/time/type milliseconds
/spatdif/time/stamp/ 123456789
SpatBASE

SpatBASE (http://redmine.spatdif.org/projects/spatdif/wiki/SpatBASE) is a rather comprehensive database for spatialization software. The motivation behind SpatBASE is to gather information of different Spatial Sound Renderer in form of a wiki. Each application has its own wiki-page and contains information such as sound rendering features, parameter definition (syntax, data range, verbal description) and system requirements. Also paper references and art projects that have been realized with that specific audio renderer are listed. On the one side with SpatBASE, we want to promote different spatialization approaches by giving them a collective web presence. On the other side, these collected information are useful for the SpatDIF development by finding commonalities and differences between renderers.
Applications already using a standardized scene-description.

- ICST Ambisonics tools

- Jamoma

- OMPrisma
Potential outcome of this meeting

• Formalization of requirements in consensus with the community
• Proposals for implementations
• More Definitions i.e. Directivity ? how to describe ?
• Storage vs. Stream Issues ?
• What is needed for the webpage of SpatDIF ?
• Searchable database structure for the SpatBASE ?
• We need scene examples -> Benchmark scenes!!!

• we need a model for decision making
Future efforts

Nils Peters will stay at IRCAM for a month this summer to research & develop strategies to store SpatDIF descriptors in files for the purpose of sharing spatial scene descriptions between spatial audio applications and across research institutes. Furthermore, several sound scenes will be created using this developed storing solution, intended to be used as reference scenes to perceptually evaluate spatial audio renderer.
Publications


Thank You

Nils Peters
Trond Lossius
Jan Schacher
Marlon Schumacher

www.spatdif.org