On Hyperstructure and Musical Structure

David C. De Roure, Don G. Cruickshank, Danius T. Michaelides,
Kevin R. Page and Mark J. Weal
Intelligence Agents Multimedia Group
Department of Electronics and Computer Science
University of Southampton
Southampton, UK
{dder,dgc,dtm,krp,mjw}@ecs.soton.ac.uk

ABSTRACT
In this paper we report on an ongoing investigation into the relationship between musical structure and hyperstructure, based on a series of open hypermedia systems research projects that have featured case studies involving musical content. We provide a general overview of the intersection between hypermedia and musical structure, drawing also on ideas from narrative structure. Through the example systems we consider techniques for building hyperstructure from musical structure and, conversely, building musical structure from hyperstructure. Additionally we describe an experiment in the sonification of hyperstructure.

Categories and Subject Descriptors
H5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia - *Architectures, Navigation;* H5.5 Sound and Music Computing - *Methodologies and techniques.*

General Terms
Design, Human Factors.

Keywords
Hyperstructure, musical structure, narrative structure, open hypermedia.

1. INTRODUCTION
A piece of music, like a work of literature or cinema, has traditionally been delivered to its audience in a linear rendition – but like these other forms it actually has rich internal structure. The field of hypertext has explored many issues relating to such structure, for example through hypertext fiction. In this paper we explore the relationship between musical structure and hyperstructure.

Many hypermedia systems incorporate musical content. Typically, the nodes of the hypermedia structures are opaque chunks of multimedia content – fragments of digital audio for example. Our goal here is to look at musical content as something which is itself a hypertext rather than a node within one.

Standards such as HyTime ('Hypermedia/Time-based Structuring Language') [24] and SMIL ('Synchronized Multimedia Integration Language') [26] provide mechanisms for integrating hypermedia and temporal media; the Amsterdam Hypermedia Model [10] extends an established model to include time and context. Our investigation in this paper explores the nature of the hyperstructure, rather than any particular encoding.

1.1 Background
Designing hypertext systems to incorporate multimedia content brings a set of challenges related to manipulating different representations and their associated storage, transmission and presentation issues. Interactive applications impose a variety of additional requirements on the systems. Our study has been conducted through a series of experimental hypermedia systems that have worked with musical content in order to study a number of these research issues.

The authors first addressed open hypermedia and audio in the set of tools described in [6,7]. In subsequent research we focused on musical content. Music, in its digital forms, has additional characteristics which favour its adoption in case studies for experimental hypermedia systems; for example it touches on similar challenges to textual content as well as those of temporal media. We have found it attractive for the following reasons:

- There are multiple representations of the same content, ranging from digital audio, to MIDI, to score representation languages. As such it is a natural candidate for hypermedia systems which separate structure from content.
- The content is structured, with different kinds of structural information provided by the different representations; e.g. MIDI describes a linear sequence of note events, while a score representation language might describe arrangements.
- It is a rich format, in that many forms of information may be bound together, perhaps through temporal synchronisation. This includes annotations that occur during the processing (composition, production, performance) of the music. The inherent structure of a piece of music is common amongst multiple instances of a particular piece.

HT'02, June 11-15, 2002, College Park, Maryland, USA.

- Legacy content is readily available, and new content is being generated all the time. There are extensive linking opportunities with associated text and media. New content production involves digital processing and automation, enabling metadata to be captured upstream in the workflow.
- Performances may be live or recorded. Live content raises challenges that are less familiar in textual studies, including on-the-fly processing of content. Live and recorded content can be interwoven.
- Storage and transmission of musical content exercises systems aspects: it may be streamed, or handled in a store-and-forward fashion; it may also be compressed and handled in the compressed domain. Quality of service issues apply.
- Applications may also be collaborative; e.g. music may be composed or performed by several musicians, perhaps geographically distributed.
- Computer music technology is a mature field. There are many software tools for working with musical content, and associated standards. Tools have been available for many years to a wide range of users.
- Music shares rights management issues with other digital content, made more complex through various additional agencies of creative input between composition and listening.

When demonstrating our systems we have encountered one common misconception: while people understand why they may wish to interact in a branching fashion with a radio news programme or documentary, music is often regarded as a purely passive medium. However, composition, performance, production and delivery of music involve interaction.

One can envisage a new class of applications that involve more active engagement. Such an experiment is described in [20] where '…hypermedia access makes the listener a participant in an evolutionary compositional process. The situation puts in question traditional roles of composer and performer'. There are surely creative opportunities here analogous to those of hyperfiction.

1.2 Overview

Since traditional music applications will not completely motivate the systems we describe, the next section considers the many aspects of working with hyperstructure in the more familiar territory of textual hypermedia systems. Section 3 then considers musical structure. Building hyperstructure from musical structure is the subject of section 4, illustrated by example systems, and section 5 considers the converse. In section 6 we consider sonification of hyperstructures. The reflections in section 7 return to the textual aspect, considering the relationship with narrative structure, and finally section 8 discusses the work and identifies areas for future study.

2. HYPERSTRUCTURE

Open hypermedia systems are characterised by hyperstructure that is separable from the multimedia content. The information that comprises the hyperstructure is first class; it is metadata, highly structured and sharable and can be manipulated independently from the content. Note that hyperstructure may be applied to static or dynamic content, and may itself be static or dynamic.

The combination of hyperstructure and content effectively provides a highly adaptable virtual document. This notion is rooted in the Xanadu model [14] in which the document (called a 'virtual file') might quote spans of characters from the original content; here that could equally be musical notes. Indeed, one might consider a piece of music formed from a set of digital audio samples to be exactly this.

Through our hypermedia tools and systems, we create hyperstructure, publish it, deliver it, reuse it, repurpose it, maintain it and discard it – a hyperstructure lifecycle. To identify the variety of aspects of working with hyperstructure, we consider three questions and outline the spectrum of possible answers:

What information is in the hyperstructure?

> By definition, the hyperstructure consists of sets of associations. A single association may have several sources, destinations, and additional data. A source or destination may be highly specific (e.g. a particular range of bytes in specific file, a tagged location, a time interval or frame number, a polygonal region in a map) or more general (e.g. a pattern to match in a set of documents, or a search query involving use of synonyms).
>
> The simplest open hypermedia link is a pair of URLs; i.e. a source and destination. Other data models for links include those used in HyTime [24], OHP [5] and FOHM [13].

Where does the hyperstructure come from?

> It may be created manually during the authoring of the content, or subsequently by annotation. For example, links might be created by adding associations using an authoring tool, textually or graphically.
>
> It may already be explicit, as when working with HTML. Implicit structure of a document can be made explicit using content-based analysis techniques; e.g. automatically identifying paragraph breaks in a text file and marking them up as HTML.
>
> It may also be derived from a richer hyperstructure by removing links (this is the 'sculptural' approach [3]) or by combining existing hyperstructures.
>
> Multiple hyperstructures may be produced to repurpose the same content for different users and applications.

When and how is the hyperstructure used?

> The hyperstructure may be used at various points in the lifecycle, for example when a document is authored, published or delivered.
>
> A published document may or may not have links in it, and they may or may not need a link service to resolve them. For example, a document may be published and delivered with inline links as in a Web browser; alternatively, it may

be used by an application which presents information to the user without links, such as an adaptive information system using a hypermedia system as a back end [22].

Hyperstructures can be managed by link services.

The use of document features, as opposed to explicit locations, supports ‘generic linking’, i.e. linking on content. Generic links provide a powerful tool for authoring (producing multiple links from which the author selects – the sculptural approach again) and working with pre-authored or dynamic content (which is then ‘linked automatically’ by the application of generic links). Although not a defining characteristic of open hypermedia, generic linking has been utilised in many open hypermedia systems [8].

3. MUSICAL STRUCTURE

Music inherently contains large amounts of structure at various levels. The basic notes and performance directions (tempo, volume, style) a composer uses to capture a piece, can all be used to describe that which we normally work with in the audio domain. In turn these can be used to derive musical structure.

Many works exhibit a musical ‘form’ (binary, ternary, variation, sonata, and suchlike – these are terms established in the western music tradition) and within these employ structured repeats and codas. Melodic themes are exposed, modified, and re-introduced, while chords often follow well-defined progressions.

While this structure can be annotated on the musical score, other representations of the same content (for example, a recording) do not explicitly carry this metadata. Some composers and musicologists use techniques such as Schenkerian analysis [19] to derive higher level structure from music, which might be based on pitch, temporal elements, or both.

Some of the higher level representations for music describe musical structure, and it is not surprising that computer languages to describe music are closely related to programming languages (for example, the language used in Section 6 is based on Lisp). Standard Music Description Language (SMDL) [25], for example, is build on HyTime and aimed at representation and exchange of musical information; it encodes a structure in which one ‘cantus’ has multiple ‘threads’, where a thread is a sequence of musical events.

We now revisit the questions from the previous section, with a view to using musical content:

What information is in the hyperstructure?

Locations in musical representations can be very specific to the representation (e.g. the use of MIDI song position pointers) but many formats can make use of some form of temporal reference – either a time code or a bar number. There are many different types of associations. The same locations in different representations of the same music may be linked – in fact these might be regarded as multiple source specifications for one association. The constituent components of a piece of music may be linked together. There may also be links to associated objects outside the music itself.

There are many possible features in musical representations that can be used for matching purposes (and hence generic linking). In section 4.2 we describe the use of melodic pitch contours as one example.

Where does the hyperstructure come from?

Manual addition of links during composition, or applied to existing musical material, requires new or enhanced tools. For example, a sequencer or notation package could be extended. Where tools are sufficiently programmable, this interface can be created using macros or a plugin. One can also envisage entirely new interfaces.

Automatic derivation is possible based on musical structure, especially when working from a high level representation. For example, a declarative description of the arrangement of a piece of music can ‘compile down’ into a hyperstructure and associated set of parts. One hyperstructure may be applicable to more than one piece of music.

When and how is the hyperstructure used?

During performance, structure can be used to play the musical sections in their predefined order, or the performer may interact with the hyperstructure; e.g. it could be used to suggest all possible steps forward based on the current context. The later also applies during composition, which can be viewed as creating a single hyperstructure out of the dense set of all valid links. In some 20th century aleatory music (also known as ‘chance music’), the order of sections is chosen at the point of composition (publishing), in other works it is left until the performance (delivery).

During production, hyperstructure can be used for rapid navigation of the work. Different arrangements and mixes can be viewed as alternative hyperstructures for the same media content.

Some music formats provide the option of including link information, such as the text track in Quicktime, tags in MP3 files or direct encoding in MIDI. Hence links can be included at publishing time and used when the music is delivered.

In the next sections we explore several mappings between musical structure and hyperstructure.

4. BUILDING HYPERSTRUCTURE FROM MUSICAL STRUCTURE

In this section we look at two examples where open hypermedia systems are used to create a navigational hyperstructure from musical structure. Both examples are applications designed to assist the user in working with multiple representations of music, as illustrated in Figure 1, and they do this by introducing hyperstructure to assist navigation within and across representations.

4.1 Continuous Link Flow

The HyStream system [16] uses a continuous link flow to annotate and navigate streamed media content. One of the original goals of this project was to apply the system in a musical environment, such as an educational application.

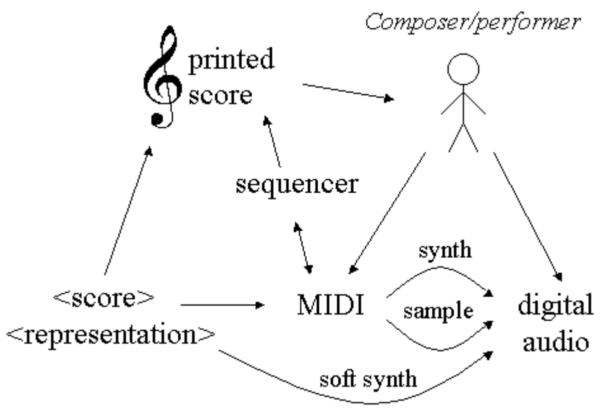

Figure 1. Multiple representations

For our musical scenario we deal with two main types of document. The first is an audio representation of the music, in this case a MIDI version although a sampled (digital audio) recording could be used, which would then be streamed. The Windows Media Player (top left hand corner of Figure 2) is used to render the audio.

The second representation of the music is the score of the piece. The scoring package (Coda Finale) outputs the manuscript as a series of image files, normally over a number of pages. Using simple image analysis to locate bar lines within the score, we create an HTML image map to demarcate the page into bars. The images that make up the score can be displayed using a standard web browser (right hand side of Figure 2).

The HyStream server holds associations in its XML linkbase between time points in the media file, and the images that represent corresponding pages in the score. It then uses a customised HTTP dialect to deliver the links to the client in a timely manner with regard to the playing media. The client displays the links in a frame below the media player (bottom left hand corner of Figure 2).

For this musical application the majority of links refer to positions in the score images. These are used to:

- Highlight the bar of music currently playing with thick under- and over-lines (bar 33 in Figure 2).
- Cue 'page turns'. When the music reaches a given point, a link is received that loads the next score page.
- Load the correct score page when an arbitrary point in the music is selected in the media player. This requires the server to restart the flow of links from the new time reference.
- Link to 'external' information sources such as information about the composer's life and work.

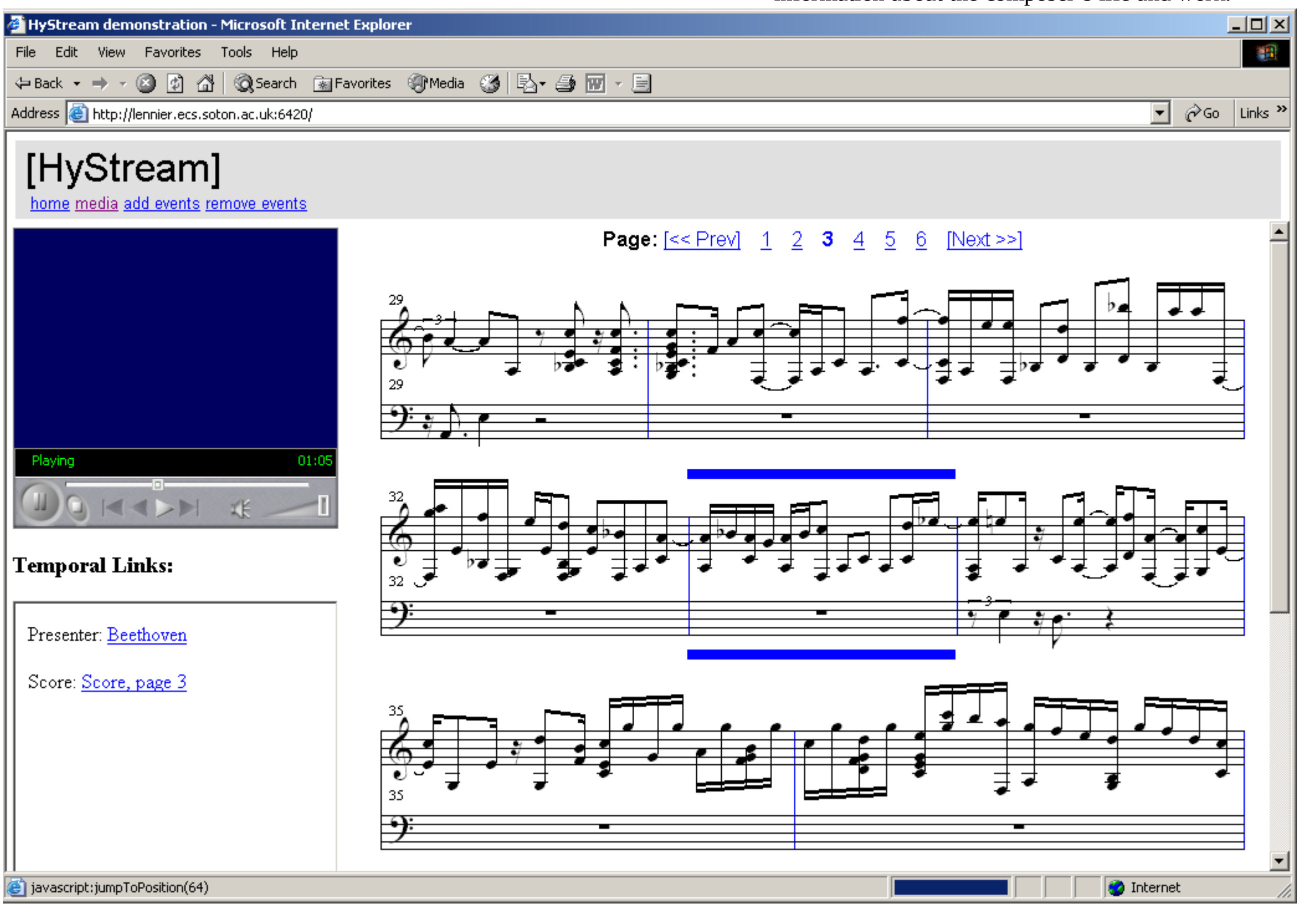

Figure 2. The HyStream client

In addition, by selecting a bar within the score window (using the image map), the corresponding link is followed, instructing the Media Player to jump to that position in the recording and resume play from that bar, also restarting the server's continuous link flow.

Because the link flow is delivered as continuous metadata, not pre-loaded information, it would be possible to use this solution for a live performance. The audio could be streamed to the client media player without any intermediate storage, but allowing a small delay for encoding. If the timing of bar changes could be retrieved from the performers, then these could be uploaded to the server and passed onto the client within the encoding delay time.

By analysing the score representation of the music and time points in the recorded representation, we have generated a navigational hyperstructure from the music's temporal structure.

HyStream has been used with a digital music archive, where it is working with stored data that may be streamed. To explore live content it is currently being used with the streamed output of a student radio station, which has associated metadata generated at the studio. It is interesting to consider its application within musical performance, where the stream(s) can be annotated with events relating to the performance. Such events could be used to provide navigation structures over recordings of the performance.

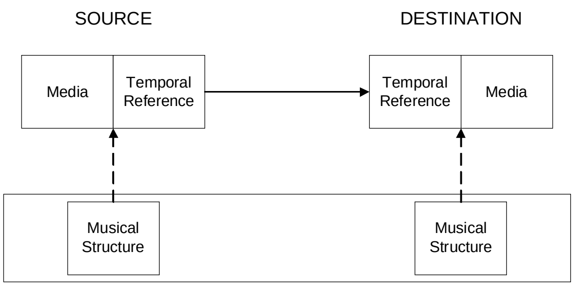

Figure 3. Source and destinations of links in the Hystream demonstrator

By analysing the score representation of the music and time points in the recorded representation, we have generated a hyperstructure from the music's temporal structure. We navigate the musical structure *through* the temporal structure of the media. For example (illustrated in Figure 3):

> A user of the system wishes to navigate to a position in the score in which they have visibly identified a particular musical structure. A link exists on the bar in the score with a source that, although implicitly representing musical structure, is actually referenced using a position on the image (there is no explicit musical structure). The link is followed; the destination is a temporal reference in the media currently playing. Again, the destination is to a reference rather than directly to the underlying musical structure.

The HyStream system is using manually authored links accompanied with links extracted from a graphical representation of the music. The next example investigates the extraction of musical structure directly from a musical representation.

4.2 Melodic Pitch Contours

The concept of generic linking, whereby links are not anchored to locations in files but rather contain patterns which match against content, was introduced in section 2. Here we explore the use of melodic pitch contours as an example of a pattern that can be matched against musical content in order to implement such a generic linking mechanism.

The use of melodic pitch contours [9] as a feature for generic linking was discussed in [7]. Subsequently the system was enhanced to work with MIDI streams, including a live MIDI input stream derived from a MIDI instrument (or a monophonic instrument played live through a MIDI converter) that is broken down into a series of contours.

A melodic pitch contour is essentially a sequence of symbols (up, down, same) describing the transitions between notes in a musical part. As a feature it abstracts away from absolute pitch information and from rhythm. A sequence of approximately fourteen symbols turns out to be effective in isolating individual songs in a large corpus [4]. Figure 3 shows a fragment of music in which each bar has the contour "UDU"; i.e. up-down-up.

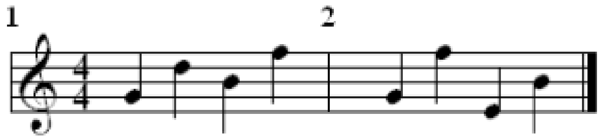

Figure 3. Bars 1 and 2 each have the contour UDU

The system architecture is show in Figure 4. The usual form of input to the retrieval engine is a standard MIDI file. It is clear how this system can be used to resolve a link that contains a contour as a destination, enabling such a link to be valid over different bodies of musical content. Consider also the case where the link contains a contour as a source anchor – such a link will also become valid when matched against content.

Users can also make live queries. To translate live data into a query format, the system buffers a certain amount of input from the client, and when the buffer contains enough data it conducts the same process as for a non-live search. Queries are made to follow on from each other by throwing away the first notes from the old query but keeping the last few and adding new input to them, creating an overlap between searches. The live MIDI data is transmitted over the network so that the instruments can be remote from the query generator and retrieval engine.

The above example uses stored content and dynamic queries. However, the content might be dynamic, and a set of generic links containing contours could be static. As the content arrives (perhaps from a performance) it is linked up on the fly using the generic links. The application to live material is intriguing – the link becomes active once the appropriate pattern is played. One interesting application of this system could be the use of the musical input to control the navigation.

The use of generic links versus absolute references (e.g. millisecond offsets into audio files, or timecodes) has a number of advantages, particularly when offsets are not robust, perhaps due to changing content.

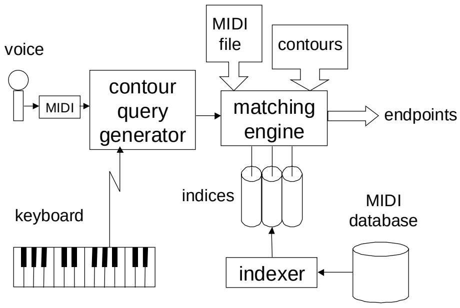

Figure 4. Melodic pitch contour system

5. BUILDING MUSICAL STRUCTURE FROM HYPERSTRUCTURE

In this section, we show hyperstructure can represent and generate musical structure. This allows us to use open hypermedia tools and link services in novel and interesting musical scenarios.

Two of these tools are the Auld Linky contextual link server [12] and the FOHM data model [13]. Auld Linky models hyperstructure as FOHM 'association' objects that relate data items together. Objects can also be annotated with context and behaviour. The presence of context attached to an object means that the object is only visible or valid if the context is matched, whereas behaviour indicates some form of action that must be performed when a data item is viewed or an association traversed.

When a client queries the server, it also passes context information, which is a set of attribute-value pairs. The server matches the query against the stored hyperstructure and filters the results with the client context; only associations and data items with matching with context are returned.

The client can maintain some state relating to the session, modified by behaviour. This means that the links available from a particular point will be different according to what has gone before.

This feature is significant for music. For example, the context defines whether it is the first or second time through a repeated section of music, and this enables the implementation of repeats and codas, as demonstrated in the next example.

5.1 Calligraphic Authoring

The majority of hypertext is created by taking a collection of text fragments and explicitly authoring links between them. The term *calligraphic hypertext* has been used to describe this process [3].

A piece of music can also be represented as a collection of fragments and a set of links that specify which fragments follow each other. A musical composition may contain several repeated fragments where the linking leads to a cyclical hyperstructure. The number of visits to a fragment can be used to break the cycle.

In Auld Linky we have modelled this using the context and behaviour attached to the hyperstructure. The context contains identifiers for the sections and associated values indicating the number of times the fragment has been played. Behaviour attached to the fragments indicates how to update the context.

While playing a fragment, the application updates the context as specified by the fragment's behaviour and the link server is queried for valid continuations. This process is repeated until no further fragments are available, resulting in a linear rendition of the hyperstructure.

Figure 5 shows a common musical form represented as a hyperstructure so as to make links explicit. The structure A, A1, A, A2, B consists of the fragments A, A1, A2 and B with the links A-A1, A1-A, A-A2, A2-B forming the transitions between them. The transitions A-A1 and A-A2 are marked as only valid when the context indicates the number of times A has been played (1 or 2 respectively).

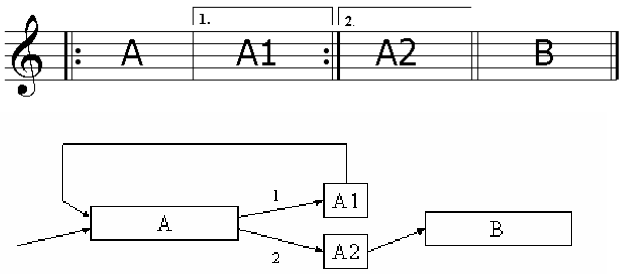

Figure 5. Structure of A,A1,A,A2,B example

This simple example only has one possible linear rendition because at no point is more than one fragment available. The performance of music represented by this type of structure need not be passive. In the more complex case where a choice of link to follow exists, the system could select one or offer the composer, performer or listener the choice.

The musical structure above can be likened to the Joyce cycle noted by Bernstein [2] as being features of Michael Joyce's Afternoon [11]; indeed the diagram has striking similarities to the map view that would be achieved using Storyspace and Guard fields to implement such a structure. Interestingly Bernstein's example includes the ballad refrain, which has obvious equivalence to the popular musical ballad.

5.2 Sculptural Authoring

Due to the nature of the reading process, relatively little has been said about conjunctive hypertext [17]. Alternatively music structure is naturally a parallel process where multiple instruments will be simultaneously playing different parts.

Sculptural hypertext provides an alternative methodology for examining musical structure, differing from calligraphic hypertext in starting with all nodes being linked, the role of the author being to remove links from the system to limit the paths for the reader. This is often done by scoping when fragments of the hypertext are available. Sculptural hypertext has been explored in textual hypertext fiction systems [3][22].

Here we present an implementation of this idea, using MIDI channels. Consider a simple MIDI representation of a song. Multiple channels represent the multiple parts and can be mapped onto a simple hypermedia structure, represented in Figure 6 using a FOHM association.

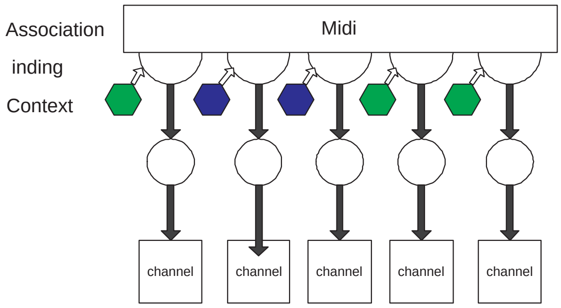

Figure 6. A representation of a MIDI file.

The channels are bound to a FOHM association. Initially, all of the channels exist as possibilities and to sculpt our musical hypertext we need to remove some of these possibilities. This is achieved by attaching a context object at each of the bindings, which specifies under what conditions the channel should be played.

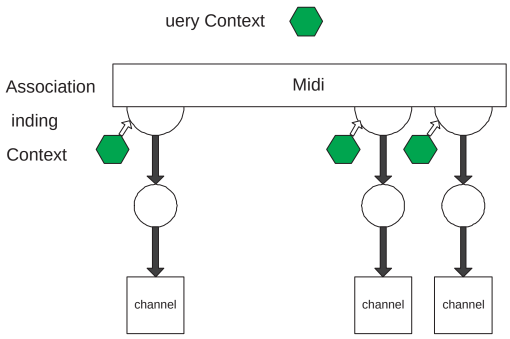

Figure 7. The pruned structure.

Upon playing the piece, the structure is extracted from Auld Linky by querying the linkserver with a context. The context is matched against the bindings and any bindings that do not match are pruned from the structure. Figure 7. Illustrates a pruned structure.

For any given context, only a subset of the channels will be appropriate to play. In our simple illustration the contexts are mutually exclusive but far more complex contexts could be imagined, taking into account the mutual compatibilities and incompatibilities between various parts.

Note that we are not suggesting that a MIDI file with multiple channels is necessarily a hypertext, but rather that it can be used to implement one. Only certain subsets of channels are designed to play simultaneously. For example, two disjoint subsets of channels might represent the same music before and after a key change, and would be discordant if played together.

6. REALISING HYPER STRUCTURE AS MUSICAL STRUCTURE

6.1 Sonification

Having explored the use of hyperstructure to navigate music, in this section we consider a role for music in assisting navigation: if hyperstructure can represent music, then here we have music representing hyperstructure.

Sonification has been explored previously in the context of the Web [1] but our approach is very different. The essential idea is the sonification of the hyperstructure, so that the user can get a preview of the hyperstructure beyond a link by listening to music generated algorithmically from that structure. The sonification tool was written to create a sonified short story for "The Bernies" competition at the ACM Hypertext 2000 conference. The story, called "The Timid Cybergoat", is a simple variation of a traditional children's story and the structure is depicted in Figure 8.

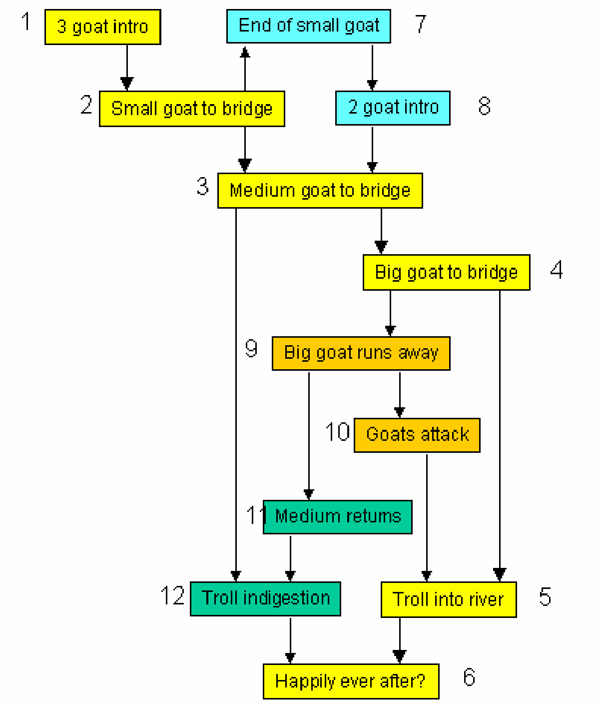

Figure 8. The hyperstructure of the Timid Cybergoat

6.2 Mapping structure to music

We have experimented with a number of approaches to converting the hyperstructure to musical structure; two are described here.

The first uses a first order Markov process. The graph is represented as a series of transition rules, with a unique note assigned to each node. The result is effectively a repeated random walk over the structure. This version, which was coded in Common Music and used the Stella editor, was used for the Bernies entry.

Subsequently we have written new tree traversal algorithms in Common Music. These perform a depth first search of the tree, with the notes assigned as before. The size, variation and repetition in the topology of the hyperstructure can be interpreted from the music, with different subjective effects from one slow traversal to repeated fast traversals. The tree for the first node of the Timid Cybergoat is shown in Figure 9, and the repetition of tree structure is visible. Note that this particular hyperstructure is acyclic; in general the traversal algorithms need to handle the situation in which a link returns to an earlier point in the story such that the traversal would repeat.

Similar effects are obtained by using drum sounds instead of notes: the harmonic relationship between notes has no significance in this particular mapping. We have also explored simultaneous notes, and the use of consonant intervals so that simpler structures sound more `harmonious'.

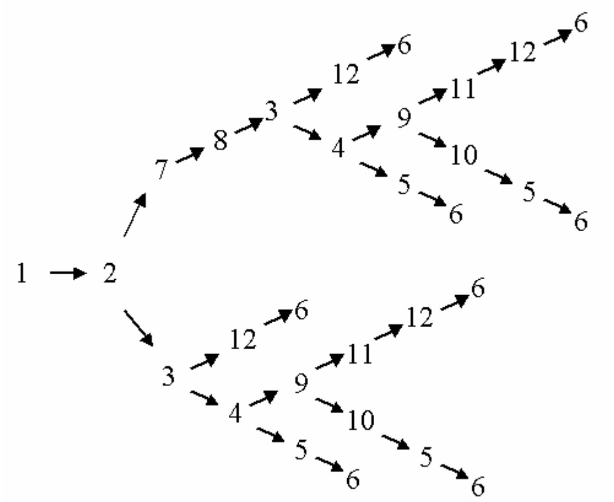

Figure 9. Tree representation of the hyperstructure

Sonification is an interface technique – it is a 'visualisation' of the hyperstructure. We explored another visualisation in a different 'Bernies' entry, in which music and links had an abstract 3D interpretation (see the cover of [23], also www.3dmidi.com).

7. REFLECTIONS ON NARRATIVE STRUCTURE

So far we have explored the relationships between musical structure and hyperstructure. In this section we return to the textual domain and present a brief discussion of the relationship with textual hyperstructures.

Musical structure can also be used as a direct reflection of narrative structure. The work of the soundtrack composer involves picking out and amplifying the emotional themes of a narrative and reflecting them in the music. A more literal interpretation might be observed in a work such as Prokofiev's Peter and the Wolf, where the voices of the characters in the story are adopted by instruments in the piece.

7.1 Patterns of Hypertext

Patterns of hypertext structure have been noted in narrative fiction [3] and similar structures can be noted in musical structure as illustrated below:

Cycle – revisiting a node to depart in a new direction.

The recalling of themes (sometimes distinctive motifs of textures) is a well-known music that exhibits cyclical characteristics. Examples might include Beethoven's 5th and 9th symphonies and Berlioz Symphonie Fantastique.

Counterpoint – two voices, interleaving themes.

Because of the parallel nature of music counterpoint (simply the independent movement of parts against one and other) has been found in most classical musical styles since Baroque and is the foundation of much harmonic theory. A more direct comparison can made to the classic concerto where a solo instrument interleaves with the main body of the orchestra.

Mirrorworld – intertextual narrative gives a contrasting perspective to the main theme.

Theme and variations presents a signature motif then applies transformations in rhythm, pitch, sequence etc while still being based on the original theme. For example Rachmaninov's Rhapsody on a Theme of Paganini, Bizet, The Women of Arles.

The Tangle – The reader is presented with a number of possible routes to follow but with little information to guide their choice.

Aleatory (chance) music presents the composer or performer with multiple possibilities (regarding sequence, tempo, pitch etc) from which they must choose often through use of a chance device (coin toss, I-Ching). In some situations the performer is confronted with a physically unplayable part forcing a choice of interpretation. For example Cage Music of Changes, Boulez, 3rd piano sonata.

Montage – several distinct writing spaces reinforce each other.

A quodlibet is a composition in which several different previously introduced tunes or melodies are presented simultaneously in a polyphonic setting. For example, the finale of Bach's Goldberg Variations, and the Tonight Quintet from Bernstein's West Side Story.

Sieves – hierarchical structures of links.

The fugue is an example of a form which presents the listener with a subject, which is then followed by derivatives of the subject and counter-subjects, all of which are related to the original through various devices; for example, the 48 fugues of Bach's Well Tempered Clavier.

Neighbourhood. – Nodes are associated by proximity.

A musical motif can be used to emphasise continuity between sections of a work. This is often used to good effect in film scores where themes can act as a signifier for either specific characters or mood. Although often melodic the ornament might be associated by timbre. Fragments may be 'close' in many musical respects; e.g. melodic similarity, harmony, question and answer phrases. There is also a notion of spatial proximity in the physical space (the stereo sound stage) and in the layout of the printed score.

Split/Join – Threads depart only to meet up again further along.

Having introduced a theme, it may be developed separately by two instruments which later return to unison/harmony. In a familiar jazz format, the band plays the 'head' and then each band member performs a solo, returning to the tune at the end.

Missing link – Suggesting the presence of a link that doesn't actually exist.

An example might be Elgar's "Enigma" Variations, in which many variations are introduced. Many believe there to be a common theme associating the variations, although it is never stated or its existence conclusively proven.

Navigational Feint – a possible change of direction is foreshadowed but can't be followed immediately.

Chord progressions are often used to build expectation of a key change only to deviate from the obvious cadence. A composer may build familiarity with a particular melodic sequence such that the listener expects one theme to follow another; when revisiting the theme the composer is able to play off of the expectations built up by instead introducing new material.

It is interesting to note that many of the structures thought innovative in hypertext form basic building blocks for compositional technique; the intrinsic parallelism of music lends itself to the exploration of multiple branches simultaneously.

8. DISCUSSION

Informed by early experiments, this paper has illustrated the explicit use of musical structure within hypermedia systems and explored how the two domains might interact and interrelate. We have illustrated a variety of relationships between musical structure and hyperstructure, drawing on work with hyperstructures in the text domain. While some practical applications are apparent, we believe that there is potential for new creative applications, perhaps challenging the perception of music as a passive medium. In principle a similar study could be made on hyperstructures in other multimedia content types.

One of the fascinating things about this study is the insights gained by mapping ideas across domains; e.g. narrative to music. It is interesting to consider the converse mapping. Reflecting on well understood musical structures may give us new insights and perspectives on the equivalent hypertext structures; in some ways, music may give us a more natural medium for exploring some of these ideas. For example, how can knowledge of hypertext patterns inform new performance techniques? Can we envisage applications working with live text?

Future work will further explore these mappings, and consider new applications and interfaces. We are particularly interested in live interactive (even collaboratve) applications, and blurring the distinction between stored and live (linking across time). For example, using the contextual link sever we can consider these ideas within systems that are sensitive to spatial context, both in a physical space and a virtual space.

9. ACKNOWLEDGMENTS

The HyStream client was developed under EPSRC project HyStream (GR/M84077/01), in collaboration with BT Exact Technologies, and the notation interface was developed by Clive Minnican. The melodic pitch contour search engine was designed and developed by Steven Blackburn, and the IPv6 streaming MIDI system by Matthew Swoboda under the EU 6INIT project (IST-1999-12383). The development of Auld Linky was supported by EPSRC projects LinkMe (GR/M25919/01) and Equator (GR/N15986/01); Equator has also supported our collaboration with Mark Bernstein. Musical content was provided from the Norman Del Mar collection, which is one of the University of Southampton Libraries Special Collections, and we are grateful to the archive staff for assistance with digitization. The 3D midi visualisation software was created by Marc Cardle, and the Timid Cybergoat story was designed by Catherine De Roure.

10. REFERENCES

[1] Barra, M., Cillo, T., De Santis, A., Matlock, T., Petrillo, U.F., Negro, A., Scarano, V. and Maglio, P.P. Personal WebMelody: Customized Sonification of Web Servers, Proceedings of the 2001 International Conference on Auditory Display, Finland.

[2] Bernstein, M., Patterns of Hypertext. in *Hypertext '98*, (Pittsburgh, PA, 1998), ACM, 21-29.

[3] Bernstein, M., Card Shark and Thespis: exotic tools for hypertext narrative. in *Hypertext 2001: Proceedings of the 12th ACM Conference on Hypertext and Hypermedia*, (Århus, Denmark, 2001), ACM, 41-50.

[4] Blackburn, S.G. and De Roure, D.C. (1998) A tool for content based navigation of music. Proceedings of ACM Multimedia '98 p.361-368.

[5] Davis, H.C., Reich, S. and Millard, D. A proposal for a common navigational hypertext protocol. Technical report, Dept. of Electronics and Computer Science, 1997. Presented at 3.5 Open Hypermedia System Working Group Meeting. Aarhus University, Denmark. September 8-11.

[6] De Roure, D.C. Blackburn, S.G., Oades, L.R., Read, J.N. and Ridgway, N. (1998) Applying Open Hypermedia to Audio. Proceedings of ACM Hypertext 98 p.285-286.

[7] De Roure, D.C. and Blackburn, S.G. (1998) Amphion: Open Hypermedia Applied to Temporal Media. Proceedings of the 4th Open Hypermedia Workshop p.27-32.

[8] Fountain, A and Hall, W and Heath, I and Davis, H (1990) Microcosm: An Open Model for Hypermedia with Dynamic Linking. Hypertext: Concepts, Systems and Applications, Proceedings of ECHT'90, Paris, November 1990 p.298-311.

[9] Ghias, A., Logan, J., Chamberlin, D., and Smith, B. C. Query by humming - musical information retrieval in an audio database, in Proc. Multimedia'95 (San Francisco, California, November 1995).

[10] Hardman, L, Bulterman, D.C.A and van Rossum, G. The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model, Communications of the ACM 37 (2), Feb 94, pp 50 - 62.

[11] Joyce, M. Afternoon, a story. Eastgate Systems, 1987.

[12] Michaelides, D.T., Millard, D.E., Weal, M.J. and Roure, D.C.D., Auld Leaky: A Contextual Open Hypermedia Link Server. in *Proceedings of the 7th Workshop on Open Hypermedia Systems*, (Aarhus, Denmark, 2001), ACM.

[13] Millard, D.M., Moreau, L., Davis, H.C. and Reich, S., FOHM: A Fundamental Open Hypertext Model for Investigating Interopability Between Hypertext Domains. in *Hypertext 2000*, (San Antonio, Texas, 2000), ACM, 93-102.

[14] Nelson, T.H. Xanalogical Structure, Needed Now More than Ever: Parallel Documents, Deep Links to Content, Deep Versioning, and Deep Re-Use. ACM Computing Surveys. Volume 31, Number 4. December 1999.

[15] van Ossenbruggen, J. and Eliens, A. Music in time-based hypermedia. In Proceedings of the ECHT'94 European Conference on Hypermedia Technologies, Technical Briefings, pages 224-227, 1994.

[16] Page, K.R., Cruickshank, D.G. and De Roure, D.C. (2001) Its About Time: Link Streams as Continuous Metadata. The Twelfth ACM Conference on Hypertext and Hypermedia (Hypertext '01) p.93-102.

[17] Rosenberg, J., And And: Conjunctive Hypertext and the Structure Acteme Juncture. in Hypertext 2001: Proceedings of the 12th ACM Conference on Hypertext and Hypermedia, (Århus, Denmark, 2001), ACM, 41-50.

[18] Selfridge-Field, E. (ed) Beyond MIDI: The Handbook of Musical Codes, 1997.

[19] Smoliar, S. A Computer Aid for Schenkerian Analysis, Computer Music Journal, Volume 4, Number 2, pp. 41 - 59, MIT Press, 1980.

[20] Tanaka, A. Musical implications of media and network infrastructures, Hypertextes hypermedias Actes de H2PTM'01, p. 241-250. October 2001.

[21] Taube, H. and Kunze, T. An HTTP Interface to Common Music. Proceedings of the 1997 International Computer Music Conference, pp. 204-207. Thessaloniki, Greece and San Francisco, CA: ICMA, 1997.

[22] Weal, M.J., Millard, D.E., Michaelides, D.T. and Roure, D.C.D., Building Narrative Structures Using Context Based Linking. in Hypertext '01: Proceedings of the Twelfth ACM conference on Hypertext, (Aarhus, Denmark, 2001), ACM, 37-38

[23] Computer Music Journal, vol. 24, no. 1, MIT Press, Spring 2000.

[24] HyTime. ISO 10744:1997 - Hypermedia/Time-based Structuring Language (HyTime), 2nd Edition.

[25] SMDL draft international standard (ISO/IEC CD 10743), July 1995. ISO/IEC JTC1/SC34.

[26] W3C Synchronized Multimedia Integration Language (SMIL 2.0), World Wide Web Consortium Recommendation, 7 August 2001.