A Multiagent system for Content Based Navigation of Music

David De Roure, Samhaa El-Beltagy, Steven Blackburn and Wendy Hall

Multimedia Research Group, Department of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, UK
+44 1703 592418

dder@ecs.soton.ac.uk

ABSTRACT

We describe the integration of content based techniques for navigation of musical information (primarily in MIDI representation) into a multiagent system for distributed information management. This exercise illustrates the application of the multiagent approach to engineering distributed multimedia information systems and raises some design issues.

Keywords

Software agents, content based navigation, pitch contours, collaborative filtering, open hypermedia, query routing.

1. INTRODUCTION

The goal of our research activity is to bring music into the distributed multimedia information space that we manage and navigate through the use of hypermedia tools. Music proves to be an excellent example of multimedia data: it exists in a variety formats, some highly structured, it can be transmitted using a store-and-forward or streaming model, it can be synchronized with other media. Hence our investigation addresses more general research issues in the application of software agents to multimedia data, with a hypermedia perspective.

Our approach is to bring together two existing research projects. The first, reported in [2], is a multiagent system for distributed information management, designed to provide navigational assistance in the organizational context; it pursues the idea that information acquired by one user navigating the information space is something that can be utilized to assist other users. The second is the 'contour engine' reported in [1], which was developed for experiments in content-based navigation of music using melodic pitch contours. Together these projects constitute our initial multiagent system for content based navigation of music.

The next section discusses the background to this work, followed in section 3 by a description of the system architecture and some of the agents. The implementation is described in section 4. We discuss general issues that have arisen through this work in section 5.

2. BACKGROUND

Our approach to distributed multimedia information management is that of *open hypermedia,* whereby information about associations (hyperlinks) between multimedia documents is stored in link databases (*linkbases*) and the documents themselves remain in their native formats. Link databases have been abstracted into a network service in work on the Distributed Link Service (DLS) [3] and the Open Hypermedia Protocol (OHP) [4]. This has been extended to incorporate linking based on features extracted from multimedia content (so-called *content based navigation* [10]) and to address temporal media. This is a powerful combination of techniques; for example, it is possible to identify arbitrary locations in arbitrary multimedia documents and find associated links in a content- and context-specific way, ending the 'tyranny of the button' [7].

The cost of this functionality is a more complex system: in addition to the documents there are linkbases to create, maintain and interrogate, and this must be achieved efficiently in a distributed environment. Increasing heterogeneity of content makes further demands on the link resolution algorithms, requiring techniques specific to content type.

We adopted a multiagent system approach as a strategy to address the engineering challenges of this increasingly complex environment [5], basing our work on the concept of *weak agency* as defined by Wooldridge and Jennings [12]. A multiagent system is composed of a group of agents that are autonomous or semiautonomous and which interact or work together, to perform some tasks or achieve some goals. The agents in such systems may either be homogeneous or heterogeneous and they may have common goals or goals that are distinct [9]. The design of individual agents within a multiagent system has the advantage of being independent of the design of other agents as long as each agent abides by an agreed upon protocol and ontology. This contributes significantly to the breakdown of complexity.

The approach was successful in the MEMOIR project [11], which brought another entity into the system: the users' *trails;* i.e. records of the documents each user has visited. Trails provide information from which associations between documents and people can be derived and used to augment navigational capabilities. Our subsequent system, reported in [2], was a multiagent system incorporating ideas from MEMOIR in a new agent framework using standard agent communication techniques (e.g. messaging based on KQML); the system also extends the work on trails to include bookmarks. Translating this into a musical context, these systems could respond to queries such as 'who else has listened to this?', 'what else did they listen to?' and 'who else is interested in music like this?'.

Finding available hypermedia links based on the content of the current location or selection (generic linking) was an innovation of the Microcosm system (discussed in [7]) and has subsequently been explored for images in the MAVIS (Multimedia Architecture for Video, Image and Sound) projects [10]. We chose melodic pitch contours (a sequence of pitch direction changes) for our work with musical content, producing a set of tools that were demonstrated at ACM Multimedia 97. Our focus on content based navigation, rather than retrieval, imposed particular requirements and resulted in a system with very fast contour lookup; the mechanisms are described in [1].

3. SYSTEM ARCHITECTURE

We focus here on the content-based part of the system, illustrated in Figure 1. The reader is referred to [2] for a description of other agents, including the organizational memory agent.

The *MIDI cache* is a repository of MIDI files and associated metadata. The files are gathered from the Internet, by a standard resource discovery agent which 'crawls' the Web for MIDI files or by agents which monitor specific MIDI archives for change. New files are notified to other agents.

Two agents work with MIDI files in the cache. The first verifies the uniqueness of files, which is currently achieved by checking both file lengths and checksums; duplicate files are removed. The second analyses the files in order to determine the nature of each MIDI channel for subsequent use in indexing and searching; this automatically derived metadata is stored.

Two databases are constructed from the MIDI files in the cache. The first is the contour database, which stores the melodic pitch contours indexed for fast access; there are two forms of contour, based on successive and alternate notes, and these are generated by the contour extractor agent which is used extensively in this system. The second is a text database based on text extracted from the MIDI files, which can also be utilized in searches.

A key feature of the contour engine is the use of 'fuzzy queries' rather than a fuzzy search algorithm. The very fast response makes it more efficient to generate multiple queries to the contour database based on the incoming query, rather than performing a search with a customized match algorithm.

Figure 1 illustrates three query sources: in example 1, the query is derived from a MIDI file and an exact match may be adequate; in example 2 the query is derived from a keyboard and multiple queries can be generated to handle common errors in this mode of input; in example 3 the query is derived from a pitch tracker and may be subject to many artifacts. The specialization of the query generation can be compared with the way in which a spelling checker takes account of common spelling errors, such as transposition of letters.

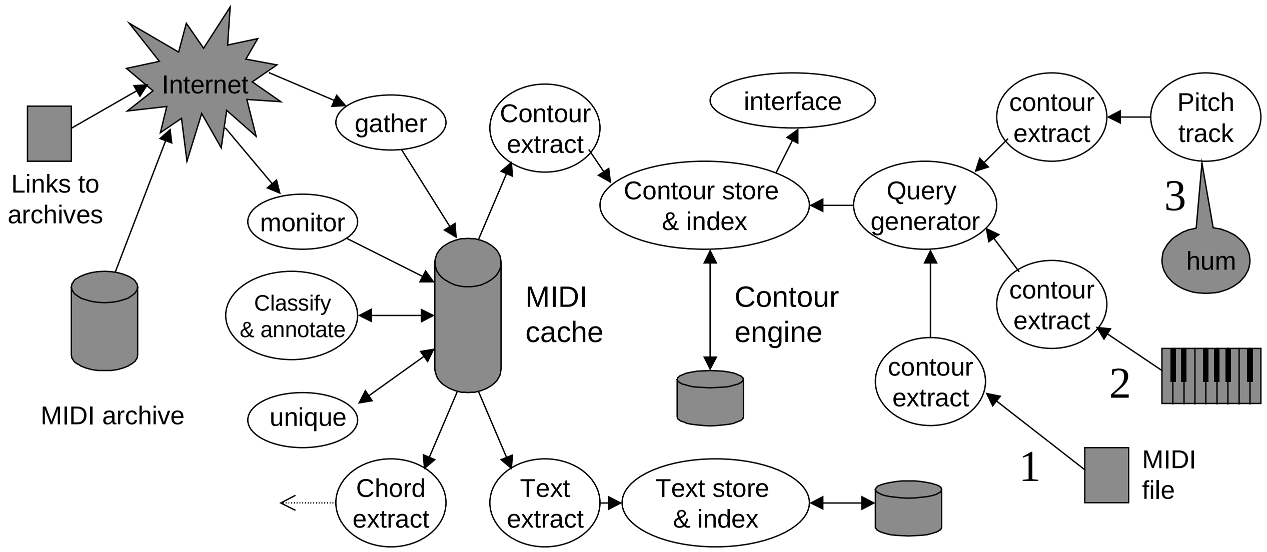

Figure 1. Agent communication in the content based navigation system

4. IMPLEMENTATION

Figure 1 illustrates flows between agents but conceals the architecture of the underlying agent framework. One of the most important agents within this framework is the facilitator agent (the agent server) that contributes to the openness and extensibility of the architecture, complying with notions of facilitation as defined by Khedro and Genesereth [8]. Mainly, it supports routing of information, discovery of information, delayed and persistent notification of messages and communication management. When agents come online for the first time, they register with the facilitator agent using the KQML register performative. They then advertise their services to the facilitator, which in turn broadcasts the advertisement to all registered agents. As soon as an agent registers, it also receives all active advertise messages that were previously sent to it by other registered agents.

The contour engine itself is 'wrapped' by the agent responsible for contour storage and access. To illustrate this and to provide a glimpse of an audio ontology, consider one of the wrappers we have implemented which makes the engine appear as a 'virtual knowledge base'. This implements a predicate called **hasContour** with arguments *mediaObject*, *contour*, *distance*, and *rank*. The media object is an object containing the URL of the MIDI file, the contour is an object containing two strings representing the primary and secondary contours (transitions between adjacent and alternate notes), the distance is the distance limit *d* for the fuzzy queries, and the rank is the position in the ordered list of matches. When the agent is queried with a contour and distance, it interrogates the contour database and returns a number of **hasContour** predicates, one for each match and each containing the appropriate media object and rank.

The MIDI channel classifier is potentially a sophisticated agent (or set of agents) and one approach would be to incorporate learning, so that the agents can be trained to recognize (for example) melody, chords and rhythm. This is the subject of future work and we currently favor a rule-based approach; meanwhile, the current implementation makes decisions based on an assumption of files compliant with the *General MIDI* conventions. The system is also being extended to support rhythm contours, again based on a three-letter alphabet (corresponding to durations of successive notes being shorter, same and longer).

This system can make use of other link services; it can also be part of a link service itself. In order to explore this and investigate the scalability of such systems, we conducted an experiment in *query routing*. In this model, one agent has information about the information held by another, and can refer a client to it directly. This requires some form of *forward knowledge*; i.e. an agent needs to be able to summarize its knowledge and pass this to other agents. We achieved this for experimental purposes by incorporating other link servers (DLS) and by splitting the contour database into two separate databases with appropriate routing.

5. DISCUSSION

Practical experience with the design and implementation raises three design issues:

The first is that in certain places the system communicates a contour within the agent communication language, but elsewhere contours are stored much in the same way as documents; in fact, transient contours are communicated and persistent ones stored. Sometimes this control *vs.* content is clear-cut, such as when an agent controls the streaming of audio data from a dedicated server. In fact it is a characteristic of working with multimedia content that the content itself might not always be readily available: e.g. sometimes an agent will be required to extract fragments at source – an argument for moving the agent to the data rather than vice versa.

The second is a related point. If we store features (and perhaps other intermediate forms) as documents, we need to maintain associations between these entities. Fortunately we have an infrastructure designed to support hypermedia, which is intended for exactly that! This suggests a model where the notion of a document is replaced by an entity consisting of a set of links, to the document content and all its associated information (for example, historical versions, alternative formats, intrinsic and extrinsic metadata).

Finally, the query routing experiment exposes another issue. The agent communication infrastructure already deals with routing messages to agents which advertise that they can deal with those messages, hence queries are routed to agents which can deal with that kind of query. Should this be extended to deal with summaries of the information that agents hold, or should that be an orthogonal mechanism? — in the current system they sit at different levels.

6. CONCLUSION

We have demonstrated a multiagent system approach to a content based navigation system. We have focused on music but believe the techniques will extend naturally to other forms of content. In particular, this research has reinforced our belief that multiagent systems are an appropriate engineering approach to distributed multimedia information management and navigation, and in particular enable us to address the increasing heterogenity and scale of these systems.

In addition to the developments discussed in section 4, planned future work includes further exploration of agents in the context of streaming media (from stored and live sources), on a QoS-enabled network and in a mobile environment.

7. ACKNOWLEDGEMENTS

This project is partially supported by EPSRC projects GR/37060 and GR/M84077. The authors are grateful to Luc Moreau, Nick Gibbins, Danius Michaelides and colleagues for establishing the audio ontology, and to Nigel Walker of BT Laboratories for assisting in the query routing experiments.

8. REFERENCES

[1] S. Blackburn and D. DeRoure, "A tool for content based navigation of music," in Proceedings of ACM Multimedia 1998, Bristol, UK, Sep. 1998, pp. 361-368.

[2] S. El-Beltagy, D. DeRoure and W. Hall, “A Multiagent system for Navigation Assistance and Information Finding”, PAAM99 - The Fourth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents. Nwana H. S. and. Ndumu D. T (Eds.), London, UK, 1999.

[3] L. Carr, D. De Roure, W. Hall and G. Hill, “The Distributed Link Service: A Tool for Publishers, Authors and Readers”. Proceedings of Fourth International World Wide Web Conference: The Web Revolution, Boston, Massachusetts, USA, December 1995, pp. 647-656.

[4] H. C. Davis, D. E. Millard, and S. Reich, N. Bouvin, K. Grønbæk, P. Nürnberg, L. Sloth, U. Wiil, K. Anderson, “Interoperability between Hypermedia Systems: The Standardisation Work of the OHSWG”, in Proceedings of Hypertext 99, Darmstadt, Germany, Feb. 1999, pp 201-202.

[5] D. De Roure, W. Hall, H Davis and J. Dale, “Agents for Distributed Multimedia Information Man-agement”. Proceedings of the First International Conference on The Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, 1996.

[6] A.Ghias, J. Logan, D. Chamberlin, and B. C. Smith. “Query by humming - musical information retrieval in an audio database”. In Proceedings of ACM Multimedia 95, San Francisco, California, November 1995.

[7] W. Hall, "Ending the tyranny of the button," *IEEE Multimedia*, vol. 1, Spring 1994, pp. 60-68.

[8] T. Khedro and M.R. Genesereth, “Facilitators: A Networked Computing Infrastructure for Distributed Software Interoperation” Presented at the Workshop on AI in Distributed Information Networks, IJCAI-95, Montreal, Quebec, 1995.

[9] V. Lesser, “Multiagent Systems: An Emerging Subdiscipline of AI”. ACM Computing Surveys, 27(3): 340-342, September 1995.

[10] P. H. Lewis, H. C. Davis, S. R. Griffiths, W. Hall, and R. J. Wilkins, "Media-based navigation with generic links," in Proceedings of the 7th ACM Conference on Hypertext, (New York), ACM Press, Mar. 1996, pp. 215-223.

[11] A. Pikrakis, T. Bitsikas, S. Sfakianakis, M. Hatzopoulos , D. DeRoure, W. Hall, S. Reich, G. Hill and M. Stairmand, “MEMOIR - Software Agents for Finding Similar Users by Trails”, PAAM98 - The Third International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents. Nwana H. S. and. Ndumu D. T (Eds.), London, UK, March 1998, pp. 453–466.

[12] M. Wooldridge, and N. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 2 (June 1995).