Music Representation – between the musician and the computer

Alan Smaill[1,3] Geraint Wiggins[2,3] Eduardo Miranda[2,3]

Abstract

In any automated tool designed for use in a musical context, the choice of representation language is crucial. The Charm system, as presented in [Wiggins *et al.*, 1993], was designed as a general purpose musical representation system. In this paper we consider how a system intended to foster the creative exploration of a musical soundworld could make use of such a representation. The ability to support multiple perspectives upon musical material is thought to be important, and we explain how this is supported. We also indicate how to incorporate automated support for the process of generalising from musical examples to a higher-level description.

1 Introduction

There is a growing interest in the use of Artificial Intelligence techniques in musical domains, from musical cognition to composition tools (see [Laske *et al.*, 1992]). In constructing AI systems, the problem of designing a suitable representation is always important. In this paper we describe an approach to music representation that we believe is appropriate for a wide range of musical systems.

In previous papers [Wiggins *et al.*, 1989, Harris *et al.*, 1991, Smaill *et al.*, 1993, Wiggins *et al.*, 1993], we have attempted to build a framework for the representation of music that would be useful for a wide range of musical tasks, from composition to analysis. Our aim has been to provide a description of a system that the user can configure to the purpose at hand; user configurability is crucial given the disparate range of activities we want to support. The system should also support the automated manipulation of music objects and structures in a way that is intuitive to the musician. Finally, we aim to remain neutral between the description of musical phenomena as processes in the physical world, and music as perceived by the musician and listener.

In this paper, we first describe properties that we believe are needed in a system to allow non-expert users to explore the process of combining musical elements, going beyond simple juxtaposition. We then review our proposal for an *abstract musical repres-*

[1]Department of Artificial Intelligence
[2]Departments of Artificial Intelligence and Music
[3]Email:{A.Smaill,G.A.Wiggins,E.Miranda}@ed.ac.uk

entation, and its configurability by an appropriate hierarchical organisation of the musical material. We indicate how these ideas can be used in constructing such exploratory musical systems. A partial implementation of the system exists, and could easily be linked up to a sound generation device, yielding a basic musical exploratorium. Systems such as Xenakis's graphical UPIC system ([Xenakis, 1992]) suggest that with appropriate technological support, musical creativity outwith conventional styles can be encouraged without needing a deep musical background. We believe that the capability to move between different underlying pitch and rhythmic perspectives would encourage such openness, and that the proposed representation provides the basis for such a system.

Such systems aim to allow the user to communicate with the system in terms as close to a natural musical vocabulary as possible, that is to say allowing the user to use something close to the vocabulary that she would use to communicate her musical ideas to fellow musicians.

To support this, we also indicate how Machine Learning techniques may be used to allow the system to infer abstract musical concepts from a set of examples.

2 A tool for musical invention

We now outline briefly what capabilities we believe should be present in a tool for musical exploration.

The user should be able to:

- enter simple musical objects (notes, motifs, ...),
- transform the objects, using some system-provided routines,
- group together objects by fiat,
- build new ways of grouping and characterising objects, and
- build new ways of transforming objects.

Of these, the last is probably the hardest to come to terms with, both for the user and for the provision of automated support. But we believe it is an important feature for a system aimed at encouraging exploration.

What sort of representation language is appropriate for such a tool?

3 Languages for music

A good knowledge representation language lets the user express knowledge of a given domain naturally and concisely, and supports an efficient reasoning regime to retrieve and manipulate the knowledge encoded. The expression should be *natural* in that the formalism used to express the knowledge should correspond closely to a description that

the user might employ spontaneously in the course of communicating with others information about the domain.

Much work has gone into the development of such formalisms, which allow a declarative reading of the encoded knowledge [Brachman and Levesque, 1985, Brachman and Levesque, 1992].

General purpose Knowledge Representation systems can be used for musical applications: for example, KL-ONE [Brachman and Schmolze, 1985] has been used as the basis for the system described in [Camurri *et al.*, 1992]. We have chosen to develop a representation system specifically for musical applications, where certain musical parameters receive special treatment.

It is notoriously hard to find a vocabulary to describe musical experience. Supposing we want to start by paying attention to some note-like events (rather than, say, looking primarily into the timbre of individual sounds). We propose to take such notes as the building blocks of our representation – yet we do not want to be committed to any particular organisation of pitches or rhythms. Our primary interest is in the music itself, either considered as a physical phenomenon, or as experienced by the listener; our interest is not (for example) the scores of the classical music tradition (which form one particular way of notating some aspects of a piece of music).

The Charm system (Common Hierarchical Abstract Representation for Music) is an attempt to free the representation of music from application- or domain-specific influence. The proposal (based on the notion of *abstract data types* from computer science) is that we need to know only some underlying properties of the pitch and time organisation for the representation to function. In different instances, we might work with a classical even-tempered semitone scale, or with a quarter-tone scale, or with some non-regular scale, or indeed with a notion of pitch where intermediate pitches are always available. Each one of these is a realisation of the *abstract* data type of pitch that we have adopted for the Charm system.

In Charm, what we require of the pitch organisation is

- a pitch value corresponding to each note, and
- a notion of *interval* between notes,

together with ideas of how these fit together. We suppose that temporal information is organised in a similar way, with points in time, and time intervals corresponding to pitches and pitch intervals respectively; there is provision too for description of timbre and dynamics.

A piece of music is then represented as a set of *events*, each of which is characterised according to pitch, time and timbre characteristics. Note that this is done without any commitment to some particular organisation of pitch (say), so that manipulation of pitches can be arranged as appropriate for the case in hand.

This provides us with a representation of the musical "surface", in the sense of Jackendoff ([Jackendoff, 1987]). Music as experienced and constructed involves of course a lot more than a set of isolated and independent events. The next representational problem is

how to deal with the enormous range of ways of thinking about music that are possible, in such a way that the power of the computer can be harnessed to musical description and invention.

4 Hierarchical structures

So far, we have described music in terms of individual notes arranged in a space of pitch and time. The performer and listener have a richer idea of music than this, and we describe a very general mechanism for the grouping of musical events that can be used both for representing salient features in some given musical experience, and as the basis for the creation of new experiences. It is widely agreed that the use of hierarchical descriptions is especially important for music representation ([Balaban, 1992, Buxton and others, 1978], so as to allow expression of local stylistic features as well as global structuring properties.

Organisation at this level is something that systems like the UPIC do not support well. Yet the organisation of music around notions of similarity and contrast is basic to musical understanding.

Charm events can be grouped by the construction of "constituents". These are arbitrary collections of "particles", that is of events (which do not have to be contiguous), or of other constituents. Each constituent has a unique name, and may be labeled with a set of first order logical formulae describing the properties of its particles or of the constituent as a whole.

The idea is that formal logic gives us a language to describe the particles of a constituent, and to state properties of the particles that justify grouping them together. For example, two groups of notes can be said to *overlap* if some note from one group and some from the other sound at the same time; this relation can be captured by a logical formula.

It is also possible to label a constituent "definitionally" — to state that a particular collection of particles should be considered as a unit (*e.g.,* a piece or motif. Thus properties without a basis in deductive logic can be represented by associating a label with a particular set of musical structures.

We thus have two ways of building up our higher-level descriptions, corresponding to the two ways in which sets of objects are commonly defined in mathematics. Suppose we have already some notes $n_1, n_2, \ldots$ and some constituents based on these notes $C_1, C_2, \ldots$ We could then form a new constituent based on logical properties of the n_i and C_i; for example:

$$D_1 = \{ x : P(x) \},$$

where P is a logical formula, gives a constituent that picks out those n_i and C_i that make P true. We can also pick out some combination explicitly:

$$D_2 = \{ n_1, n_5, C_2, C_3 \}$$

Because of the use of logical formulae in the constituent specification, Charm allows the expression of complex relationships between different musical objects. It is also im-

portant that these relationships can not only be expressed, but that automated techniques are available to manipulate such representations, and to construct objects with the required properties. (This is, of course, the basis of Logic Programming (see [Kowalski, 1979]).)

To take a simple example, the following fragment of Webern can be analysed in several ways.

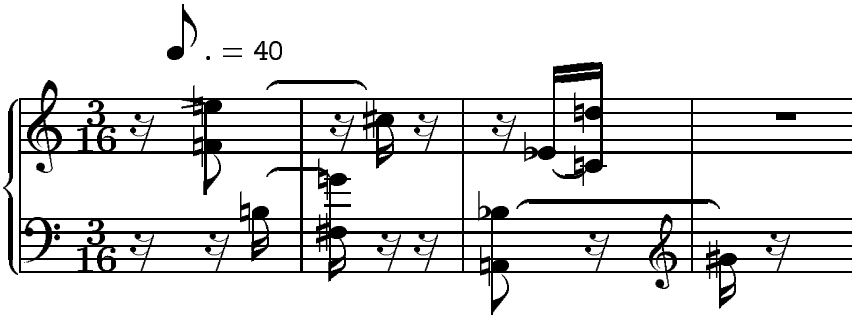

Figure 1: Webern, Op 27, bars 1-4

```
є( e00, ⟨f,♮, 4⟩, 1/4, 1/2, 10/1, ...)
є( e01, ⟨e,♮, 5⟩, 1/4, 1/2, 10/1, ...)
є( e02, ⟨b,♮, 3⟩, 1/2, 1/4, 10/1, ...)
є( e03, ⟨f,♯,3⟩, 3/4, 1/4, 10/1, ...)
є( e04, ⟨g,♮, 4⟩, 3/4, 1/4, 10/1, ...)
є( e05, ⟨c,♯,5⟩, 1/1, 1/4, 10/1, ...)
є( e06, ⟨a,♮, 2⟩, 3/2, 1/2, 10/1, ...)
є( e07, ⟨b,♭, 3⟩, 3/2, 1/2, 10/1, ...)
є( e08, ⟨e,♭, 4⟩, 7/4, 1/4, 10/1, ...)
є( e09, ⟨c,♮, 4⟩, 4/2, 1/4, 10/1, ...)
є( e10, ⟨d,♮, 5⟩, 4/2, 1/4, 10/1, ...)
є( e11, ⟨g,♯,4⟩, 9/4, 1/4, 10/1, ...)
```

Figure 2: Webern, Op 27, bars 1-3

The figures 1 and 2 show a conventional score representation, and an instantiation of Charm's basic event representation for the same fragment. This takes the form a set of note descriptions, where the arguments shown are successively identifier, pitch, start time, duration, and dynamic.

The passage can be thought of (and heard) in several ways – in terms of texture, or pitch symmetry, or temporal symmetry, or as a series. Our contention is that the ambiguity that allows this passage to be heard in many ways is central to this musical experience, and that our representation language should support such multiple viewpoints upon a single musical surface (to use the term of [Jackendoff, 1987]).

Many people have stressed the importance of multiple viewpoints in computational descriptions of music (for example, [Minsky, 1981, Lerdahl and Jackendoff, 1983,

Ebcioglu, 1988, Holland and Elsom-Cook, 1990, Conklin and Witten, 1991]). We would like not only to be able to translate easily between different representations, but even more – to have a single representation in which the multiple viewpoints can be expressed and compared.

The constituent mechanism allows just this. For example, various constituent are shown in figure 3.

Constituent c00 is an enumerated constituent that we have chosen to call *subject*. Constituent c01 contains the same notes, but satisfies the logical specification of a *series*. c03-c06 satisfy the logical specification for *chord*, and so on.

κ(c00, _ , subject, { e00 e01 e02 e03 e04 e05 e06 e07 e08 e09 e10 e11 }, _)
κ(c01, ⟨series,{}⟩, _ , { e00 e01 e02 e03 e04 e05 e06 e07 e08 e09 e10 e11 }, _)
κ(c03, ⟨chord,{}⟩, _ , { e00, e01 } , _)
κ(c04, ⟨chord,{}⟩, _ , { e03, e04 } , _)
κ(c05, ⟨chord,{}⟩, _ , { e06, e07 } , _)
κ(c06, ⟨chord,{}⟩, _ , { e09, e11 } , _)
κ(c07, ⟨alternation,{}⟩, _ , { c03, e02, c04, e05, c05, e08, c06, e11 } , _)
κ(c08, ⟨triple,{}⟩, _ , { e00, e01, e02 } , _)
κ(c09, ⟨triple,{}⟩, _ , { e03, e04, e05 } , _)
κ(c10, ⟨triple,{}⟩, _ , { e06, e07, e08 } , _)
κ(c11, ⟨triple,{}⟩, _ , { e09, e10, e11 } , _)

Figure 3: Constituents for Webern Op 27

This brief description is intended to show that this approach allows the flexibility of multiple representations of the musical material. Such flexibility should subsequently allow high-level manipulation of the musical material in ways that make sense to the musician, who will have tailored the system to her way of thinking in the course of collecting together salient groupings, and indicating which logical descriptions are relevant.

5 Using the representation

We now show how the capabilities set out in section 2 can be built on top of this representation system.

- Simple musical objects can be built up from a repertoire of provided ingredients, preferably using a visual interface like the system UPIC, as developed by Xenakis. At this point, the higher level structuring properties are not in play, and UPIC has shown the effectiveness of this approach at this level.
- Simple transformations (repetition, augmentation, transposition, inversions, etc) are easily written in terms of the underlying abstract representation and provided to the user.
- Collecting objects together is a matter of simply indicating which among the full set of objects under consideration are to be taken. Again, mouse clicking on a

graphical representation is an obvious approach.

- Building descriptions of new groupings, and new ways of transforming objects, are two faces of the same problem in this approach. However, we do not necessarily expect the user to write logical specifications directly. A more intuitive approach would allow a mixture of two sorts of user input, as follows.
  - On the one hand, the user can ask the system to generalise from a set of examples – "here are three motifs – give me a description of a class of motifs that they come from." This is a problem in Machine Learning, and there are several well-known techniques available. We discuss this further in section 6.
  - On the other hand, the user should be able to require that parts of a full logical specification hold – for example, Boolean combinations of previously defined relations should be made available to the user.

These facilities together allow in principle the construction of musical objects with great flexibility, as is possible in systems that incorporate the computational power of a full programming language ((*e.g.* [Vercoe, 1991, Pope and others, 1992]). We believe it could also provide a good intermediary between the manipulative power of the computer and the intuition of the musician.

6 Assistance in concept formation

In this section we discuss a possible role for inductive machine learning in systems intended to aid musical invention. We will focus here on the modelling of a particular aspect of human cognition which is believed to play an important role in musical creativity: the generalisation of perceptual attributes. By this we mean the process by which the listener, when confronted with a series of sounds, instead of hearing each sound as unique, tries to find common features, perhaps of timbre, or pitch, or rhythm.

For the particular example we will describe, the object of attention is the *timbre* or tonal quality of particular sounds. This has not so far been integrated into the Charm representation. However, the example has been implemented in the context of a system we have built for musical exploration (of timbre), and the general approach applies equally to a system based on Charm.

We introduce below the fundamentals of inductive machine learning followed by the presentation of a example inductive learning engine as well as a brief concluding remark.

6.1 Inductive learning fundamentals

Inductive learning is to do with making generalisations, or classificatory rules, out of a collection of examples given in a training set. In our case the training set is constituted of sounds described by means of their perceptual attributes.

Broadly speaking, the problem of learning concepts from examples can be formalised as follows: let $\mathcal{U}$ be the universal set of objects, – that is, all of the objects that the

learner may encounter. There is in principle no limitation on the size of U. A concept C is formalised from a subset of objects in U. That is to say that the elements of a certain subset of objects in U share certain features (or properties) that can be formalised by a concept C. We say that the formalisation of a concept C defines a class C of objects in U. To learn a concept C means to learn to recognise objects in the class C. In other words, once C is learned, the system is able, for any object O in U, to recognise whether O belongs to class C or not.

Let U be the set of all sound events able to be produced by a certain digital instrument. Suppose C corresponds to a certain class of sounds in U. Once C is learned, the system can decide whether any sound event SE from U is in C.

The task of generalising a class C from a training set can now be stated as follows: given a training set S of examples, find a description F expressed in an appropriate language, such that for all objects X, if X is a positive example in training set S then X satisfies F, otherwise X does not satisfy F. As a result of learning, F is the "system's understanding" of the class C.

Inductive learning in our case is aimed at learning how to recognise novel sounds with respect to C, that is, sounds contained in U but not contained in the training set S. Consequently, the learned concept description F should be more general than the training elements themselves.

6.2 The example inductive machine learning engine

We have implemented a machine learning engine which uses two inductive learning algorithms: the ISCD (for Induction of the Shortest Concept Description) and the IDT (for Induction of Decision Trees). Detailed algorithm specifications are not within the scope of this paper, and can be found in [Bratko, 1990, Miranda, 1992]. In this paper, we describe how the first method functions.

We have implemented a timbre generation program which builds up sounds according to a user specification, which involves specifying values for attributes in a relatively high-level language, for example *vibrato, openness, attack, smoothness* ...(see [Miranda, 1992, Miranda *et al.*, 1993, Miranda, 1994] for a discussion of the meaning of these attributes). Each of these is implemented in the lower-level terms of the sound synthesis algorithm. The user may, however, be interested in describing sounds with a different vocabulary. Having designed a number of sounds (by experimenting with the values of the attributes available), the user can then specify that some number of these sounds have the quality the user is looking for, say being an "open vowel". The learning system will attempt to discover what combinations from the given attributes are important in differentiating between the sounds in this set that are and are not "open vowels" according to the user's classification.

The result of learning here is a set of rules which characterises the sounds of the training set that are positive examples of the concept in question. The main requirement of ISCD is that the rule set exactly matches the positive examples. Such a rule set is said to be complete and correct: complete because it includes all the positive examples and correct because it includes no negative examples.

Learning in ISCD is viewed as a search among possible rules with the objective of minimising the number of attributes in it. In other words it searches for the shortest concept description. Because of the high combinatorial complexity of this search, the ISCD resorts to a heuristic of scoring functions.

An example ISCD rule, when looking for a description for "open vowel" on the basis of some examples, is as follows

A sound event is "open vowel" if
it has "normal vibrato",
and "high openness".

(Recall that "normal vibrato" and "high openness" are terms from our given sound description language.) No matter how many attributes an "open vowel" has in the training set, according to the above ISCD rule the most relevant attributes for this sound class are "vibrato = normal" and "openness = high". "Most relevant" here means what is most important to distinguish an "open vowel" from other sound classes.

This algorithm has been used to construct a number of similar generalisations, as described in [Miranda, 1994]. Once the concept has been learned, the user may use the description directly in specifying new sounds, different from those that were originally picked out as typical of the new sound description.

6.3 Concluding remark

Our main reason for inducing rules about sounds is that the computer now can aid the user to explore among possible alternatives to accomplish a goal, namely in this case to design a certain sound. The user can ask the computer to "play something that sounds similar to a bell sound" or even "play a kind of dull sound", for example. In these cases the engine will find out rules for deducing which attributes are relevant for synthesising a bell-like sound or a sound with a dull colour attribute and which values they should take.

As for knowledge representation, we believe that supporting multiple perspectives upon learned musical material is important in the case of timbre, just as it is for the notions discussed earlier. This explains why we use more than one inductive learning algorithm in our machine learning engine.

The Machine Learning approach thus allows us to include logical descriptions of more abstract musical concepts, without requiring the user to formulate such descriptions explicitly, by generalising from examples.

7 Summary

We have indicated how a suitable representation language can play a central role in automated systems aimed at the encouragement of musical creativity and the exploration of the possibilities inherent in musical material, here considered at the level of notes

and above. Similar considerations apply to organisation of timbre and "non-musical" sounds, though our underlying basic representation is not aimed at that sort of exploration. We have shown how the automated construction of musical descriptions from examples can be carried out.

At this stage, our conclusions can only be speculative. While we have implemented some of the representational background required, the worth of this approach can only be properly evaluated after a system along these lines has been built and used in a controlled way. We hope to be able to do just this in the future.

References

[Balaban, 1992] M. Balaban. Music structures: Interleaving the temporal and hierarchical aspects in music. In O. Laske, M. Balaban, and K. Ebcioglu, editors, *Understanding Music with AI – Perspectives on Music Cognition*, pages 110–39. MIT Press, Cambridge, MA, 1992.

[Brachman and Levesque, 1985] R.J. Brachman and H.J. Levesque, editors. *Readings in Knowledge Representation.* Morgan Kaufmann, California, 1985.

[Brachman and Levesque, 1992] R.J. Brachman and H. Levesque. *Knowledge Representation.* MIT Press, London, 1992.

[Brachman and Schmolze, 1985] R.J. Brachman and J.G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[Bratko, 1990] I. Bratko. *Prolog Programming for Artificial Intelligence.* Addison Wesley, 1990.

[Buxton and others, 1978] W. Buxton et al. The use of hierarchy and instance in a data structure for computer music. *Computer Music Journal*, 2:10–20, 1978.

[Camurri *et al.*, 1992] A. Camurri, M. Frixione, C. Innocenti, and R. Zaccaria. A model of representation and communication of music and multimedia knowledge. In B. Neumann, editor, *Tenth European Conference on Artificial Intelligence*, pages 164–8, Vienna, 1992. John Wiley and Sons, Chichester, England.

[Conklin and Witten, 1991] D. Conklin and I. H. Witten. Prediction and entropy of music. Technical Report 91/457/41, Department of Computer Science, University of Calgary, Alberta, Canada, December 1991.

[Ebcioglu, 1988] K. Ebcioglu. An expert system for harmonizing four-part chorales. *Computer Music Journal*, 12, 1988.

[Harris *et al.*, 1991] M. Harris, A. Smaill, and G. Wiggins. Representing music symbolically. In *IX Colloquio di Informatica Musicale*, Genoa, Italy, 1991. Also available from Edinburgh as DAI Research Paper No. 562.

[Holland and Elsom-Cook, 1990] S. Holland and M. Elsom-Cook. Architecture of a knowledge-based music tutor. In M. Elsom-Cook, editor, *Guided Discovery Tutoring*, London, 1990. Paul Chapman Publishing Ltd.

[Jackendoff, 1987] R. Jackendoff. *Consciousness and the computational mind.* MIT Press, Cambridge, MA., 1987.

[Kowalski, 1979] R. Kowalski. *Logic for Problem Solving.* Artificial Intelligence Series. North Holland, 1979.

[Laske *et al.*, 1992] O. Laske, M. Balaban, and K. Ebcioglu. *Understanding Music with AI – Perspectives on Music Cognition.* MIT Press, Cambridge, MA., 1992.

[Lerdahl and Jackendoff, 1983] F. Lerdahl and R.S. Jackendoff. *A Generative Theory of Tonal Music.* The MIT Press, Cambridge, MA., 1983.

[Minsky, 1981] M. Minsky. Music, mind and meaning. *Computer Music Journal*, 5(3):28–44, 1981.

[Miranda *et al.*, 1993] E.R. Miranda, A. Smaill, and P. Nelson. A knowledge-based approach for the design of intelligent synthesisers. In *Proceedings of the X Reunion Nacional de Intelligencia Artificial*, Mexico City, 1993.

[Miranda, 1992] E.R. Miranda. From symbols to sound: Artificial intelligence investigation of sound synthesis. Research Paper 640, Dept. of Artificial Intelligence, Edinburgh, 1992.

[Miranda, 1994] E.R. Miranda. ARTIST - an AI-aided working environment for the design of intelligent sound synthesis device. *Languages of Design*, 1994. To appear.

[Pope and others, 1992] S. T. Pope et al. The Smallmusic Object Kernel: A music representation, description language, and interchange format. Anonymous ftp from ccrma-ftp.stanford.edu:/pub/st80, 1992. N.B. draft only.

[Smaill *et al.*, 1993] A. Smaill, G. Wiggins, and M. Harris. Hierarchical music representation for analysis and composition. *Computers and the Humanities*, 27:7–17, 1993. Also available from Edinburgh as DAI Research Paper No. 511.

[Vercoe, 1991] B. Vercoe. *The Csound Reference Manual.* Addison-Wesley Publishing Company, Cambridge, MA, 1991.

[Wiggins *et al.*, 1989] G. Wiggins, M. Harris, and A. Smaill. Representing music for analysis and composition. In M. Balaban, K. Ebcioglu, O. Laske, C. Lischka, and L. Sorisio, editors, *Proceedings of the 2nd IJCAI AI/Music Workshop*, pages 63–71, Detroit, Michigan, 1989. Also available from Edinburgh as DAI Research Paper No. 504.

[Wiggins *et al.*, 1993] G. Wiggins, E. Miranda, A. Smaill, and M. Harris. A framework for the evaluation of music representation systems. *Computer Music Journal*, 17(3):31–42, 1993. Machine Tongues series, number XVII; Also available from Edinburgh as DAI Research Paper No. 658.

[Xenakis, 1992] I. Xenakis. *Formalized Music: Thought and Mathematics in Composition.* Pendragon Press, Stuyvesant, New York, 1992.