Perspectives: a declarative bias mechanism for case retrieval *

Josep Lluís Arcos Ramon López de Mántaras

IIIA, Artificial Intelligence Research Institute

CSIC, Spanish Council for Scientific Research

Campus UAB, 08193 Bellaterra, Catalonia, Spain.

{arcos,mantaras}@iiia.csic.es

Abstract

The aim of this paper is to present a mechanism, called *perspectives*, to describe declarative biases for case retrieval in structured representations of cases. Our approach is based on the observation that, in complex tasks, the identification of the relevant aspects for retrieval in a given situation may involve the use of knowledge intensive methods. This identification process requires dynamical decisions about the relevant aspects of a problem and usually forces to consider non predefined retrieval indexes in the memory of cases. Declarative biases provide a flexible way of constructing dynamical perspectives for retrieval in the memory of cases. We have implemented the notion of perspectives in a reflective object-centered representation language, called Noos, based on feature terms. Finally, we have used perspectives as declarative biases for retrieval in the *Saxex* application, a complex real-world case-based reasoning system for generating expressive performances of melodies based on examples of human performances that are represented as structured cases.

This paper is being submitted as scientific paper.

1 Introduction

The research on complex representations of cases is motivated by the construction of CBR systems in complex real-world domains. Structured representations

*The research reported on this paper has been supported by the Spanish project *SMASH* TIC96-1038-C04-01. Research by J.L. Arcos was also supported by a CSIC fellowship. This work has benefited from the comments and discussions with Enric Plaza.

of cases based on the notion of objects and relations among them, usually implemented as graph structures, allow a flexible and higher expressive power that attribute-value representations.

Structured representations of cases offers the capability of treating subparts of cases as full-fledged cases: a new problem can be solved using subparts of multiple cases retrieved from the system's memory. On the other hand, structured representations of cases increase the complexity of retrieval mechanisms and requires the development of new retrieval techniques supporting the complex representations of cases.

In our research work we have described *feature terms* as a formalization of structured representation of complex cases [6, 16]. We have developed Noos, a reflective object-centered representation language designed to support knowledge modeling of problem solving and learning based on feature terms [4, 5]. In [16] a similitude measure based on a preference ordering among cases was presented. The use of inductive learning techniques using feature term representations is also investigated in [7].

The aim of this paper is to present a mechanism, called *perspectives*, to describe declarative biases for case retrieval in structured representations of cases. Our approach is based on the observation that, in complex tasks, the identification of the relevant aspects for retrieval in a given situation may involve the use of knowledge intensive methods. This identification process requires dynamical decisions about the relevant aspects of a problem and usually forces to consider non predefined retrieval indexes in the memory of cases. Declarative biases provide a flexible way of constructing dynamical perspectives for retrieval in the memory of cases. We have implemented the notion of perspectives in Noos.

We use perspectives as declarative biases for retrieval in the *Saxex* application, a complex real-world case-based reasoning system for generating expressive performances of melodies based on examples of human performances that are represented as structured cases.

The organization of this paper is as follows. In Section 2 we present the notion of feature terms and their use in the Noos representation language. Section 3 describes *perspectives* as a declarative bias mechanism for retrieval. Section 4 shows the use of perspectives in the *Saxex* application. Section 5 discusses related work. Finally, in Section 6 we present the conclusions.

2 Feature Terms

Feature terms are record-like data structures embodying a collection of *features*. The difference between feature terms and first order terms is the following: a first order term, e. g. $f(x, g(x, y), z)$, can be formally described as a tree and a fixed tree traversal order—in other words, variables are identified by position. The intuition behind a feature term is that it can be described as a labeled graph—in other words, variables are identified by name (regardless of order or position). This difference allows to represent partial knowledge.

For instance, a sequence of two notes where the first one is a *C5* with a quarter duration (noted as Q) followed by a *G4* with an eighth duration (noted as E) is described using the feature term representation as follows:

$$X : Note \begin{bmatrix} pitch & \doteq X_1 : C5 \\ duration & \doteq X_2 : Q \\ next & \doteq Y : Note \begin{bmatrix} pitch & \doteq Y_1 : G4 \\ duration & \doteq Y_2 : E \\ previous & \doteq X \end{bmatrix} \end{bmatrix}$$

We use the common dot notation for field selection (for instance $X.pitch \doteq X_1 : C5$).

Feature terms have a correspondence to labeled graphs representation. For instance, Figure 1 shows the description of a musical score containing two main interlaced sequences: the melody as a sequence of notes and the harmonization as a sequence of chords.

Our approach to formalize feature terms is related to the research based on *ψ-terms* [2, 9] that proposes formalisms to model object-oriented programming constructs. We describe the signature Σ of feature terms as the tuple $\langle \mathcal{S}, \mathcal{F}, \leq \rangle$ such that:

Figure 1: Partial browse of the score for the 'All of me' ballad. Features are represented as thin boxes, dots indicate not expanded terms, and gray boxes express references to existing terms.

- $\mathcal{S}$ is a set of *sort symbols* including $\bot, \top$;
- $\mathcal{F}$ is a set of *feature symbols*;
- $\leq$ is a decidable partial order on $\mathcal{S}$ such that $\bot$ is the least element and $\top$ is the greatest element.

We define an interpretation $\mathcal{I}$ over the signature $\langle \mathcal{S}, \mathcal{F}, \leq \rangle$ as the structure

$$\mathcal{I} = \langle \mathcal{D}^{\mathcal{I}}, (s^{\mathcal{I}})_{s \in \mathcal{S}}, (f^{\mathcal{I}})_{f \in \mathcal{F}} \rangle$$

such that:

- $\mathcal{D}^{\mathcal{I}}$ is a non-empty set, called *domain* of $\mathcal{I}$ (or, universe);
- for each symbol s in $\mathcal{S}$, $s^{\mathcal{I}}$ is a subset of the domain; in particular, $\top^{\mathcal{I}} = \mathcal{D}^{\mathcal{I}}$ and $\bot^{\mathcal{I}} = \emptyset$;
- for each feature f in $\mathcal{F}$, $f^{\mathcal{I}}$ is a total unary function $f^{\mathcal{I}} : \mathcal{D}^{\mathcal{I}} \mapsto \mathcal{P}(\mathcal{D}^{\mathcal{I}})$. When the mapping is not defined it is assumed to have value $\top$.

Given the signature Σ and a set $\mathcal{V}$ of variables, we define formally *feature terms* as follows:

Definition 1 *A feature term ψ is an expression of the form:*

$$\psi \quad ::= \quad X : s\, [f_1 \doteq \Psi_1 \cdots f_n \doteq \Psi_n]$$

where X is a variable in $\mathcal{V}$, s is a sort in $\mathcal{S}$, $f_1, \cdots, f_n$ are features in $\mathcal{F}$, $n \geq 0$, and each Ψ_i is either a feature term or a set of feature terms.

Note that when $n = 0$ we are defining only a sorted variable ($X : s$). We call the variable X in the above feature term the *root* of ψ (noted $Root(\psi) = X$), and say that X is *sorted* by the sort s (noted $Sort(X) = s$) and has features $f_1, \cdots, f_n$. The set of variables and the set of features ocurring in ψ are noted respectively as $\mathcal{V}_\psi$ and $\mathcal{F}_\psi$.

A feature term is a syntactic expression that denotes sets of elements in some appropriate domain of interpretation ($[\![\psi]\!]^\mathcal{I} \subset \mathcal{D}^\mathcal{I}$). Thus, given the previously defined interpretation $\mathcal{I}$, the denotation $[\![\psi]\!]^\mathcal{I}$ of a feature term ψ, under a valuation $\alpha : \mathcal{V} \mapsto \mathcal{D}^\mathcal{I}$ is given inductively by:

$$[\![\psi]\!]^\mathcal{I} = [\![X : s[f_1 \doteq \psi_1 \cdots f_n \doteq \psi_n]]\!]^\mathcal{I} = \{\alpha(X)\} \cap s^\mathcal{I} \bigcap_{1 \leq i \leq n} (f_i^\mathcal{I})^{-1}([\![\psi_i]\!]^\mathcal{I})$$

where $f^{-1}(S)$, when f is a function and S is a set, stands for $\{x | \exists S' \supset S$ such that $f(x) = S'\}$; i.e., denotes the set of all elements whose images by f contains at least S.

Using this semantical interpretation of feature terms, it is legitimate to establish an order relation between terms. Given two terms ψ and ψ', we will be interested in determine when $[\![\psi]\!]^\mathcal{I} \subset [\![\psi']\!]^\mathcal{I}$.

2.1 Subsumption

We have just seen that the semantical interpretation of feature terms allows to define an ordering relation among feature descriptions. We call this ordering relation as *subsumption*. The intuitive meaning of subsumption is that of *informational ordering*. We say that a feature term ψ_1 subsumes another feature term ψ_2 ($\psi_1 \sqsubseteq \psi_2$) when all information in ψ_1 is also contained in ψ_2. Formally,

Definition 2 (Subsumption)

Given two feature terms ψ and ψ', ψ subsumes ψ', $\psi \sqsubseteq \psi'$, if there is a total mapping function $\upsilon : \mathcal{V}_\psi \to \mathcal{V}_{\psi'}$ such that :

1. $v(Root(\psi)) = Root(\psi')$,

and $\forall x \in \mathcal{V}_\psi$

2. $Sort(x) \leq Sort(v(x))$,

3. for every $f_i \in \mathcal{F}$ *such that* $x.f_i \doteq \Psi_i$ *is defined, we have that* $v(x).f_i \doteq \Psi'_i$ *is also defined,*

4. $\forall \psi_k \in \Psi_i\ \exists \psi'_k \in \Psi'_i$ *such that* $v(Root(\psi_k)) = Root(\psi'_k)$, *and*

5. $v(Root(\psi_k)) \neq v(Root(\psi_j))$ *when* $k \neq j$.

For instance, the following feature term is a partial description of a sequence of two notes that subsumes all the sequences of two notes where the first one is a *C5* and the following a *G4* (specifically subsumes the previous showed example):

$$X : Note \begin{bmatrix} pitch \doteq X_1 : C5 \\ next \doteq Y : Note \begin{bmatrix} pitch & \doteq Y_1 : G4 \\ previous & \doteq X \end{bmatrix} \end{bmatrix}$$

This notion of subsumption is the basis of the retrieval mechanism in our approach. Specifically, the set of retrieval methods provided in Noos is based on the lattice generated by the subsumption ordering.

2.2 The Noos Language

Noos is a reflective object-centered representation language designed to support knowledge modeling of problem solving and learning based on feature terms. Noos is based on the task/method decomposition principle and the analysis of knowledge requirements for methods —and it is related to knowledge modeling frameworks like KADS [21] or ComMet [19].

Problem solving in Noos is considered as the construction of an *episodic model*. The view of "problem solving as modeling" is that problem solving is the construction of an episodic model from problem data and problem solving knowledge. A clear and explicit separation between tasks, methods, and domain knowledge permits a dynamical link between a given problem, tasks, and methods as well as a dynamical choice of a suitable method to achieve a task in a given resolution context : a 'task' applies a 'method' on a 'episode' (described using domain knowledge and problem data). Thus, an episodic model gathers the knowledge pieces used for solving a specific problem. Once a problem is solved, Noos automatically memorizes (stores and indexes) the episodic model that has been built. *Episodic memory* is the (accessible and retrievable) collection of episodic models of the problems that a system has solved. The memorization of episodic models is the basic building block for integrating learning, and specifically CBR, in Noos.

Noos incorporates *preferences* to model decision making about sets of alternatives present in domain knowledge and problem solving knowledge. For instance, preference knowledge can be used to model criteria for ranking some precedent cases over other precedent cases for a task in a specific situation.

3 Perspectives

The goal of the retrieval task in CBR is to search for similar precedents from the memory of cases. Our goal is to retrieve structured cases using domain specific knowledge expressed as complex relations among objects. The identification of the relevant aspects in complex situations requires the use of knowledge-intensive methods. These relevant aspects constitute the base for the search in the memory of cases. We claim that the use of declarative biases in the identification phase provides a clear and flexible way to express retrieval mechanisms in complex-real applications.

The view of feature terms as partial descriptions allows the representation of declarative biases also as feature terms in a natural way. The declarative biases are interpreted as syntactic patterns. *Perspectives* are the way to construct, from these syntactic patterns, partial descriptions of the current problem embodying the aspects considered as relevant. These partial descriptions are used as retrieval patterns for searching similar cases in the lattice of feature terms. The intuition behind perspectives is shown in Figure 2.

There are two possibilities for constructing perspectives. The first option is via a syntactic pattern using unsorted variables in some feature values. For

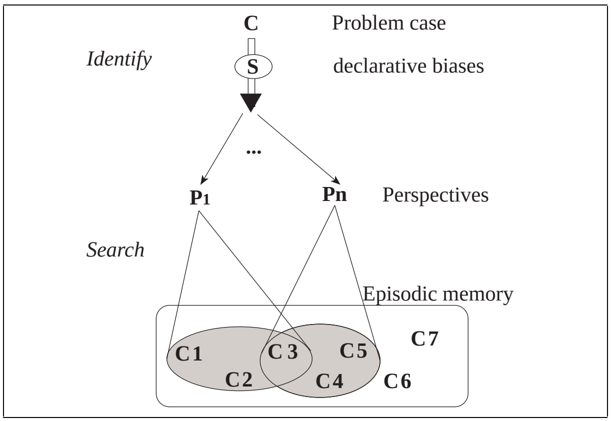

Figure 2: Using perspectives in the Retrieval task. First, given a problem case C and using syntactic patterns S as declarative biases, the identification phase determines the relevant aspects building perspectives $P_1 \cdots P_n$. Then, perspectives are used to search precedent cases in the episodic memory.

instance, in order to declare as relevant aspects of a note its duration and its metrical strength on the melody, we will use the following syntactic pattern,

$$X : Note \begin{bmatrix} duration & \doteq & Y \\ metrical\text{-}strength & \doteq & Z \end{bmatrix}$$

Then, the application of this bias to note *note_121* illustrated in Figure 1 (the theme 'All of me'), for instance, will construct the following perspective,

$$X : Note \begin{bmatrix} duration & \doteq & Y : Q \\ metrical\text{-}strength & \doteq & Z : extremely\text{-}high \end{bmatrix}$$

that in turn will be used for retrieval obtaining, as a result, the set of note precedents from the memory of cases with Q (quarter duration) and a *extremely-high* metrical strength.

The second way to build a perspective is to use a syntactic pattern where the relevance of some features is declared using variables of features. This alternative allows to identify the roles of features and terms in the structure. For instance, in order to declare as relevant the role that a given note X plays in the structure of the cognition model of musical understanding of Narmour's theory (see Section 4 for more details) we will use the following syntactic pattern where feature variables are noted with the \$ symbol,

$$X : \mathit{Note}\left[\mathit{belongs\text{-}to} \doteq Y : \mathit{N\text{-}structure}\left[\$f \doteq X\right]\right]$$

Specifically, in our example where the note *note_121* is the *first-note* of a melodic process structure, according to Narmour's theory, the following perspective will be constructed,

$$X : \mathit{Note}\left[\mathit{belongs\text{-}to} \doteq Y : \mathit{Process}\left[\mathit{first\text{-}note} \doteq X\right]\right]$$

Finally, using this perspective for retrieval we obtain all the notes playing the same role that the note problem from the memory of cases (i.e. first notes of melodic process structures).

Formally, the signature of syntactic patterns is an extension of the signature Σ of feature terms that incorporates a set of feature variables $\mathcal{L}$ ($\langle \mathcal{S}, \mathcal{F} \cup \mathcal{L}, \leq \rangle$). Thus, syntactic patterns are expressed as second order feature terms as follows:

Definition 3 *A syntactic pattern ω is an expression of the form:*

$$\omega \quad ::= \quad X : s\,[f_1 \doteq \Omega_1 \cdots f_n \doteq \Omega_n]$$

where X is a variable in $\mathcal{V}$, s is a sort in $\mathcal{S}$, $f_1, \cdots, f_n$ are features in $\mathcal{F} \cup \mathcal{L}$, $n \geq 0$, and each Ω_i is either a syntactic pattern or a set of syntactic patterns.

Given the previous definition of syntactic patterns, we define formally a perspective P as follows,

Definition 4 (Perspective)
Given a problem case C and a declarative bias defined by means of a syntactic pattern S, a perspective *of C is defined as a feature term P such that there is a total bijective function $\beta : \mathcal{V}_P \rightarrow \mathcal{V}_S$, a total mapping function $\delta : \mathcal{V}_S \rightarrow \mathcal{V}_C$, and an instantiation function $\rho : \mathcal{F}_S \rightarrow \mathcal{F}_T$ satisfying:*

1. $\rho(f) = f \qquad \forall f \in \mathcal{F}$

2. $\beta(\mathit{Root}(P)) = \mathit{Root}(S)$, $\delta(\mathit{Root}(S)) = \mathit{Root}(C)$

and $\forall x \in \mathcal{V}_P$

3. $\mathit{Sort}(\beta(x)) \leq \mathit{Sort}(\delta(\beta(x)))$, *4. $Sort(x) = Sort(\delta(\beta(x)))$,*

5. for every $f_i \in \mathcal{F}$ such that $x.f_i \doteq \Psi_i$ is defined, we have that

(a) $\exists f_j : \rho(f_j) = f_i$,

(b) both $\beta(t).f_j \doteq \Psi'_i$ and $\delta(\beta(x)).f_i \doteq \Psi''_i$ have to be defined, and $\forall \psi_k \in \Psi_i$:

(c) $\exists \psi'_k \in \Psi'_i$ such that $\beta(Root(\psi_k)) = Root(\psi'_k)$

(d) $\exists \psi''_k \in \Psi''_i$ such that $\delta(\beta(Root(\psi_k))) = Root(\psi''_k)$.

Remark that a perspective P is constructed as a partial description of a problem case C. In other words, this implies that $P \sqsubseteq C$. Another important remark is that several perspectives satisfying the definition can be obtained. This implies that the implementation of the perspectives mechanism has to provide a way to obtain all of them (for instance, by providing a backtracking mechanism).

4 Using perspectives for generating expressive performances of melodies

Saxex [3] is a case-based reasoning system for generating expressive performances of melodies based on examples of human performances. *Saxex* incorporates background musical knowledge based on Narmour's implication/realization model [15] and Lerdahl and Jackendoff's generative theory of tonal music (GTTM) [12]. These theories of musical perception and musical understanding are the basis of the computational model of musical knowledge of the system.

We study the issue of musical expression in the context of tenor saxophone interpretations. We have done several recordings of a tenor sax performer playing several Jazz standard ballads with different degrees of expressiveness, including an (almost) inexpressive interpretation of each piece. These recordings are analyzed, using SMS spectral modeling techniques [18], in order to extract basic information related to the expressive parameters. The set of extracted parameters together with the scores of the pieces constitute the set of structured cases of the case-based system. From this set of cases and using similarity criteria based on background

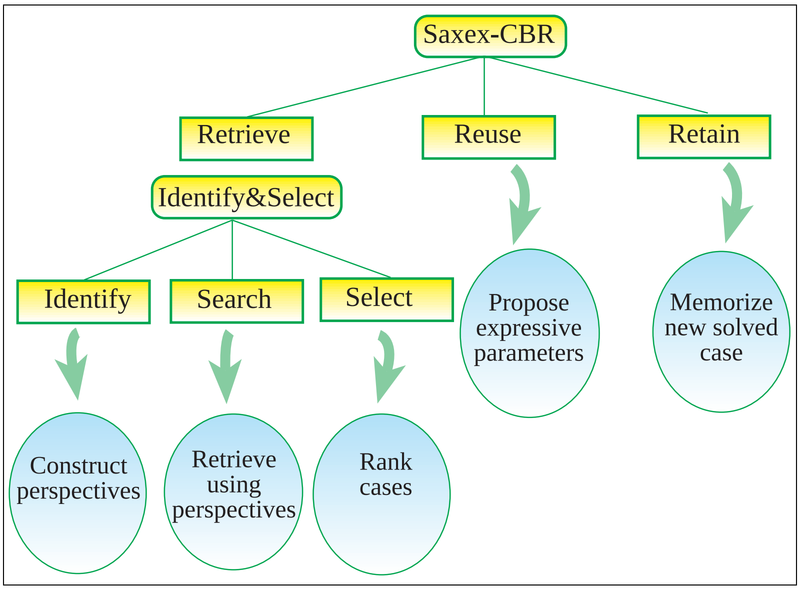

Figure 3: Task decomposition of the *Saxex* CBR method.

musical knowledge expressed as perspectives, the system infers a set of possible expressive transformations for a given piece. Finally, using the SMS synthesis procedure and the set of inferred transformations, *Saxex* generates new expressive interpretations of the same jazz ballads as well as of other similar melodies.

The problem solving method developed in *Saxex* follows the usual subtask decomposition of CBR methods described in [1]: *retrieve, reuse, and retain* (see Figure 3). The overall picture of the subtask decomposition is the follows:

- *Retrieve*: The goal of the retrieve task is to choose the set of notes (cases) most similar to the current problem. This task is decomposed in three subtasks:
  - *Identify*: The goal of this task is to build retrieval perspectives using two alternative biases. The first bias uses Narmour's implication/realization structures. The second bias uses Lerdahl and Jackendoff's generative theory.
  - *Search*: The goal of this second task is to search cases in the case

memory using Noos retrieval methods and previously constructed perspectives.

  - *Select*: The goal of the select task is to rank the retrieved cases using Noos preference methods. The preference methods use criteria such as similarity in duration of notes, harmonic stability, or melodic directions.

- *Reuse*: the goal of the reuse task is to choose a set of expressive transformations to be applied in the current problem from the set of more similar cases. The first criterion used is to adapt the transformations of the most similar case. When several cases are considered equally similar, the majority rule is used. Finally, when previous criteria are not sufficient, all the cases are considered equally possible alternatives and one of them is selected randomly.

- *Retain*: the incorporation of the new solved problem to the memory of cases is performed automatically in Noos. All solved problems will be available for the reasoning process in future problems.

Once we have described the overall picture of the CBR method, let us now explain in more detail the role of perspectives in the retrieval subtask in *Saxex*. The background musical knowledge incorporated in the system is the basis for the construction of perspectives for retrieval.

The first bias applied is based on the Implication/Realization (IR) theory of Narmour. IR propose a theory of cognition of melodies based on eight basic structures. These structures characterize patterns of melodic implications that constitute the basic units of the listener perception. Other parameters such as metric, duration, and rhythmic patterns emphasize or inhibit the perception of these melodic implications. The use of the IR model provides a bias based on the structure of the melodic surface. The example of the note's role in a process structure described in the previous section is an example of a bias used in *Saxex* following Narmour's theory.

The second bias used is based on Lerdahl and Jackendoff's generative theory of tonal music (GTTM). GTTM offers an alternative approach to understanding melodies based on a hierarchical structure of musical cognition. GTTM proposes four types of hierarchical structures associated with a piece. This structural approach provides the system with a complementary view of determining relevance biases. An example of a bias based on the GTTM theory is the use of the metrical importance of a note (see the first example in previous section).

Combining these two biases we have performed two sets of experiments. The first set of experiments consisted in using examples of expressive performances of some phrases of a piece in order to generate expressive performances of other phrases of the same piece. This group of experiments has revealed that *Saxex* identifies clearly the relevant cases even though the new phrases introduce small variations of phrases existing in the memory of cases. The second set of experiments consisted in using examples of expressive performances of some pieces in order to generate expressive performances of other pieces. This second group of experiments has revealed that the use of perspectives allows to identify situations such as long notes, ascending or descending melodic lines, etc. Such situations are also usually identified by a human performer.

As a final remark, we want to emphasize that the final output of *Saxex* are sound files containing expressive performances resulting from applying adequate expressive transformations to the situations identified and retrieved by the use of perspectives. This capability offers a simple way to test the solutions proposed by the system by just listening to the output sound files and the results obtained are very promising[1].

5 Related work

Other works on structure-based representations are CAPLAN [14], focusing on case-based planning; the research around the FABEL project [10], and the use of parallel techniques in CaPER [11, 17]. There is a growing interest on structured

[1]see <http://www.iiia.csic.es/Projects/music/Saxex> for a sample of sound example.

representation due to the higher representation power that is required in complex real-world applications.

Another related work is the stratified case-based reasoning technique presented in [8]. Their approach is to use a hierarchical representation of cases in order to reduce the complexity associated to rich case representations.

The use of CBR techniques in music applications has also been explored in Macedo et al work [13] applying CBR techniques to music composition using a tree-like structured representation of cases. Previous work addressing the issue of learning to generate expressive performances based on examples is that of Widmer [20], who uses explanation-based techniques to learn rules for dynamics and rubato in the context of a MIDI electronic piano.

6 Conclusions

We have shown that declarative biases provide flexible ways of constructing dynamical perspectives to guide the retrieval process in a memory of structured phrases. The notion of perspectives has been implemented in the object-centered representation language Noos based on feature terms. The practical feasibility of our approach has been shown in the context of a complex real-world CBR system for successfully generating expressive performances of melodies. The evaluation of the output sound files gives compelling evidence for the validity of our approach.

References

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AICom*, 7(1):39–59, 1994.

[2] H. Aït-Kaci and A. Podelski. Towards a meaning of LIFE. *J. Logic Programming*, 16:195–234, 1993.

[3] J. L. Arcos. Saxex: un sistema de raonament basat en casos per a l'expressivitat musical. Master's thesis, Institut Universitari de l'Audiovisual. Universitat Pompeu Fabra, 1996.

[4] J. L. Arcos and E. Plaza. Integration of learning into a knowledge modelling framework. In Luc Steels, Guss Schreiber, and Walter Van de Velde, editors, *A Future for Knowledge Acquisition*, number 867 in Lecture Notes in Artificial Intelligence, pages 355–373. Springer-Verlag, 1994.

[5] J. L. Arcos and E. Plaza. Inference and reflection in the object-centered representation language Noos. *Journal of Future Generation Comp. Systems*, 12:173–188, 1996.

[6] J. L. Arcos and E. Plaza. Noos: an integrated framework for problem solving and learning. In *Knowledge Engineering: Methods and Languages*, 1997.

[7] E. Armengol and E. Plaza. Induction of feature terms with INDIE. In M. van Someren and G. Widmer, editors, *Machine Learning: ECML-97*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 1997.

[8] L. K. Branting and D. W. Aha. Stratified case-based reasoning: Reusing hierarchical problem solving episodes. In *IJCAI-95*, pages 384–390, 1995.

[9] B. Carpenter. *The Logic of typed Feature Structures*. Tracts in theoretical Computer Science. Cambridge University Press, Cambridge, UK, 1992.

[10] F. Gebhardt. Methods and systems for case retrieval exploiting the case structure. Technical report, GMD, 1995. FABEL-Report N.39.

[11] B.P. Kettler, J.A. Hendler, W.A. Andersen, and M.P. Evett. Massively parallel support for case-based planning. *IEEE Expert*, 9:8–14, 1994.

[12] F. Lerdahl and R. Jackendoff. An overview of hierarchical structure in music. In Stephan M. Schwanaver and David A. Levitt, editors, *Machine Models of Music*. 1993. Reproduced from Music Perception.

[13] L. Macedo, F.C. Pereira, C. Grilo, and A. Cardoso. Plans as structured networks of hierarchically and temporally related case pieces. In *Third European Workshop on Case-Based Reasoning*, Lecture Notes in Artificial Intelligence, pages 234–248. Springer Verlag, 1996.

[14] H. Muñoz-Avila and J. Huellen. Retrieving cases in structured domains by using goal dependences. In Manuela Veloso and Agnar Aamodt, editors, *Case-Based Reasoning, ICCBR-95*, number 1010 in Lecture Notes in Artificial Intelligence, pages 241–252. Springer-Verlag, 1995.

[15] E. Narmour. *The Analysis and cognition of basic melodic structures : the implication-realization model*. University of Chicago Press, 1990.

[16] E. Plaza. Cases as terms: A feature term approach to the structured representation of cases. In Manuela Veloso and Agnar Aamodt, editors, *Case-Based Reasoning, ICCBR-95*, number 1010 in Lecture Notes in Artificial Intelligence, pages 265–276. Springer-Verlag, 1995.

[17] K.E. Sanders, B.P. Kettler, and J.A. Hendler. The case for structured-based representations. Technical report, University of Maryland, 1995.

[18] X. Serra. Musical sound modelling with sinusoids plus noise. In G. De Poli, A. Picialli, S. T. Pope, and C. Roads, editors, *Musical Signal Processing*. Swets and Zeitlinger Publishers, 1996. (in Press).

[19] L. Steels. Components of expertise. *AI Magazine*, 11(2):28–49, 1990.

[20] G. Widmer. Learning expressive performance: The structure-level approach. *Journal of New Music Research*, 25 (2):179–205, 1996.

[21] B. Wielinga, W. van de Velde, G. Schreiber, and H. Akkermans. Towards a unification of knowledge modelling approaches. In J. M. David, J. P. Krivine, and R. Simmons, editors, *Second generation Expert Systems*, pages 299–335. Springer Verlag, 1993.