The Piano Tutor: An Intelligent Teaching Assistant

Peter Capell Dr. Roger Dannenberg Dr. Annabelle Joseph
Robert Joseph David Pugh Dr. Marta Sanchez

Carnegie Mellon University
Pittsburgh, PA 15213, USA.

October 3, 1988

Abstract

The Piano Tutor is a tutoring system designed to provide piano instruction to beginning piano students with limited supervision by a teacher. In order to provide effective piano instruction, the Piano Tutor must be flexible enough to adjust its teaching strategy to the student and knowledgeable enough to correctly remediate any error the student makes. In addition, the Piano Tutor must be fast enough to keep the student interested. This is done by using two different rule-based systems. The first uses a fixed set of rules to select the next lesson to be taught. The second uses rules stored with the lesson to analyze a student's performance and do remediation.

1 Introduction

The Piano Tutor began as an effort to construct an intelligent computer-based system for giving piano instruction to college students. Two relatively new technologies were to be used. Bloch and Dannenberg's work on musical accompaniment by computer [1] makes it possible to follow a student's performance on the piano[1]. Interactive video-disks make it possible to store presentations and play them for the student as part of a lesson. Combining these technologies with intelligent teaching software allows the Piano Tutor to analyze a student's performance and call up lessons for remediation or additional instruction.

An interesting aspect of the Piano Tutor is the problem domain. Unlike other tutors that deal with cognitive tasks (Anderson's geometry tutor [2], Clancey's GUIDON [3]), the Piano Tutor must be able to handle performance errors (e.g. the student knew the correct key, but missed it by accident) as well as cognitive errors. An added complication is that we do not have a model for how people learn to play the piano. Fortunately, it seems that the best remediation for both types of errors is similar (see Section 4), and we believe an effective tutor for piano instruction can be obtained without resorting to cognitive modeling.

Another tutor which works in the domain of psychomotor skills is the CPR Tutor [4]. This system combines an instrumented mannequin with interactive video disk to provide CPR instruction to the general public. The CPR Tutor uses a fixed sequence of lessons – presentations followed by requests for the student to perform an action (such as clearing the airway). If the student fails, then the student is given a presentation to help him or her correct the error and is allowed to try

[1] The Piano Tutor is attached to a piano keyboard/synthesizer.

again. If the student continues to make the error, the CPR Tutor stops and requests assistance from an instructor. The primary difference between the CPR Tutor and the Piano Tutor is that the Piano Tutor maintains a model of the student's abilities and uses the model both to guide lesson selection (see Section 3.4) and remediation (see Section 4).

1.1 Traditional piano instruction

Traditionally, piano instruction consists of lessons with a piano teacher followed by independent practice by the student. During the lessons, the student plays pieces that he or she has practiced. The teacher evaluates the performance and offers technical and musical suggestions. In the event of serious errors, the teacher can interrupt the performance and correct the problem. Intervention by the teacher can take several forms, from a verbal suggestion to a demonstration of the correct technique.

While this method of piano instruction works, the weak link is the independent practice. Quite often, a student will either develop habits which must be "unlearned" or discover that he or she needs more guidance to render an acceptable performance. Both of these problems can be avoided by providing some supervision of the student during independent practice. Unfortunately, it is impractical for the teacher to provide this supervision.

1.2 Piano instruction using the Piano Tutor

The Piano Tutor is designed to assume the role of the piano teacher between formal lessons – following the student's performance, pointing out and remediating errors, selecting lessons, and keeping a careful record of the student's performance. A session with the Piano Tutor will typically consist of several lessons. Each lesson will present the material associated with the lesson and test the student by having him or her play a piece of music. The student's performance is then analyzed and the appropriate remediation is provided, if needed. Each lesson will normally take less than five minutes to finish, and the student can end the session at any time.

The Piano Tutor also has a passive mode in which the student picks the music to play and the Piano Tutor follows the student's performance and points out errors, but does not keep any record or provide remediation. The rationale for the passive mode is simple: the student needs the chance to experiment without feeling like Big Brother[5] is watching.

The Piano Tutor does not replace the human teacher. Instead, it aids the teacher by handling the "mechanical" aspects of piano playing. A teacher is still needed to answer questions and correct problems which are beyond the scope of the Piano Tutor (such as poor posture). The net effect of using the Piano Tutor is that both the teacher's and student's time is used more effectively.

2 Design of the Piano Tutor

One of the design requirements for the Piano Tutor is that it must be able to work with a wide variety of students. We decided that the easiest way to accomplish this was to use short, simple lessons (see Section 3.3) and a skill-based student model (see Section 3.2).

Lessons have prerequisites and objectives which are expressed in terms of skills. Lessons have prerequisites (a list of skills and a range of proficiencies for each skill) and objectives (a list of skills and a new proficiencies). Often several different lessons will teach a skill to a particular proficiency – each lesson representing a different teaching strategy. Normally, more than one lesson will have all of its prerequisites satisfied and be eligible for teaching. When this happens, a rule-based system is used to select the "best" lesson (see Section 3.4).

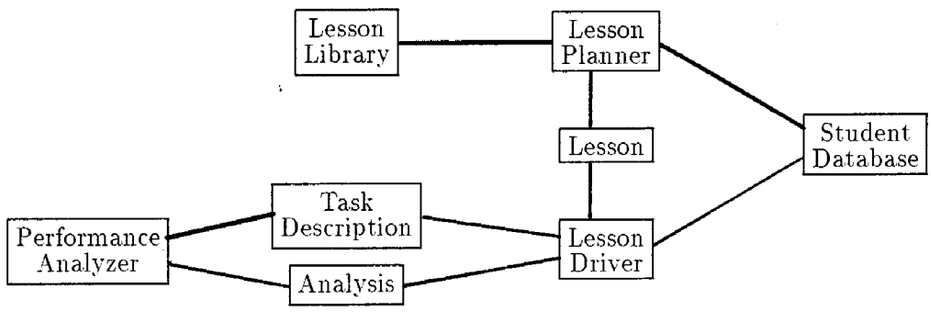

Figure 1: The Piano Tutor architecture

A second design requirement for the Piano Tutor is that it must operate in real-time during a student's performance. This implies fast decision-making which we believe can only be achieved using a procedural style of programming. However, much of the work of the Piano Tutor involves the complex decision making typically associated with searching and rule-based programming. To cope with these seemingly incompatible constraints, we use a two-level strategy in which lesson selection (see Section 3.4) and remediation (see Section 4) are carried out using relatively slow but powerful expert system programming paradigms, and performance analysis (see Section 3.1) is carried out using much faster procedural programs. This two-level approach give us fast response when needed and a structure that supports complex decision-making when response time is less critical. This is in contrast to intelligent tutoring systems that attempt to track each student action using a cognitive model.

Analyzing a student's performance to give intelligent feedback is another problem. We came to the conclusion that it was impossible to do intelligent analysis using a set of "global" remediation rules. Instead, remediation rules (see Section 4) are written for each lesson. These rules, in turn, decide what type of error has been made. The actual remediation is done by error handlers (see Section 4.1), which may either by "global" (for simple, common errors) or "local" (for errors unique to the lesson).

Associating the remediation rules with each lesson has several advantages. In particular, remediation in the Piano Tutor often works on the principal that an error with several possible causes is the result of a failure to learn the material presented in the current lesson. Since the remediation rules are written for each lesson, it is easy to incorporate the knowledge of what material is taught by a lesson into its remediation rules.

3 The Piano Tutor architecture

Figure 1 gives an overview of the Piano Tutor prototype. In brief, the Piano Tutor works as follows:

1. The Lesson Planner selects, using information stored in the Student Database, a lesson from the Lesson Library and passes it to the Lesson Driver.

2. The Lesson Driver presents any new material associated with the lesson to the student, and then passes a task description to the Performance Analyzer.

3. The Performance Analyzer follows the student's performance (using the information contained in the task description). When the student finishes the performance, an analysis is returned to the Lesson Driver.

4. The Lesson Driver then, using the performance analysis and the rules contained in the lesson, provides remedial instruction, and updates the Student Database.

Each of the different components in the Piano Tutor will be described in the following Sections.

3.1 The Performance Analyzer

The Performance Analyzer is the "real-time" element of the Piano Tutor. The Performance Analyzer uses a variation of the dynamic programming technique for identifying longest common subsequences to match a student's performance to a stored musical score [1]. This method was originally developed to allow a computer to synchronize an accompaniment to a performance by following a score, but a byproduct of the algorithm is a list of right notes, wrong notes, extra notes and omitted notes. This information forms the basis for analysis of student performances. In addition, the score-following algorithm is used to pace musical accompaniment and "turn" music pages by updating the computer display.

The task description passed to the Performance Analyzer contains the score the student should play and the accompaniment, if any. The accompaniment can take a variety of forms, from the simple "tick-tock" of a metronome to a multi-timbral synthesized "orchestra". Accompaniment can alert the student to pitch or rhythm errors without intervention by the Piano Tutor. Also, it is fun to play an accompanied piece and the promise of accompaniment can provide additional incentives for the student to practice.

Once the student has finished, the Performance Analyzer returns an error log – essentially a list of pitch, starting time, duration, tempo[2] and velocity[3] errors for each note. Currently, there is no mechanism for having the Performance Analyzer interrupt the student in the middle of a performance (if, for example, the student makes more than a certain number of pitch errors). The error log forms the basis for remediation (see Section 4). The Performance Analyzer is "robust," in that the student can make several errors (including added, skipped or incorrect notes) and the errors listed in the error log will correspond to the errors the student made.

3.2 The Student Database

The Student Database holds all of the Piano Tutor's knowledge about the student. This knowledge is stored in two separate structures: a list of skills, called the Student Model, and record of the student's and the Piano Tutor's interactions, called the Student History. Together, these two data structures store all of the information used to select lessons and guide remediation.

The Student Model The student is modeled as a list of skills and an estimated proficiency in each skill. The primary purpose of the Student Model is to guide lesson selection (see Section 3.4) and control remediation (see Section 4). Examples of skills are:

- Time-4/4
  Proficiency at maintaining a consistent 4/4 tempo.

[2] Failing to maintain a consistent tempo.

- Symbol-1/2Note
  Proficiency at recognizing and playing half notes.
- Repeat-RH
  Proficiency at playing sequences of identical notes using the right hand.
- Skip-RH
  Proficiency at playing sequences of non-consecutive notes using the right hand.

Most other tutoring systems which use a skill-based student model (e.g. Goldstein's genetic graph [6] or Heines and O'Shea's ReGIS tutor [7]) use "binary" skills – either the skill is known or it is not known (or, in the ReGIS tutor, the system does not know whether the student knows the skill). In the Piano Tutor, skills have an estimated proficiency ranging between 0 (minimum) and 10 (maximum) or "?" (unknown). All skills are initialized to unknown when a student starts using the Piano Tutor. The proficiency estimates change as a result of passing lessons and/or remediation.

The Student History The Student History is a record of the student's interaction with the Piano Tutor. Every action taken by the Piano Tutor (e.g. updating the Student Model, giving a presentation, failing a lesson, etc.) is automatically recorded. In addition, error handling routines (see Section 4.1) can make entries into the Student History as needed. The primary purpose of the Student History is to provide a convenient mechanism for detecting repeated errors across lessons.

3.3 The Lesson Library

A lesson has three parts:

1. Prerequisites.
   This is a list of skills and a range of proficiencies for each skill (which can include unknown).
2. Objectives.
   This is a list of skills and a new estimate of proficiency for each skill. If the student passes the lesson then the Student Model is updated based on the lesson's objectives.
3. Contents.
   This is a set of rules which decide which set of presentations and performances to give to the student. These rules tend to be very simple and their primary use is to prevent repeating presentations and performances when a student repeats a lesson.

From a student's perspective, each lesson consists of a series of presentations and performances[4]. Presentations are designed to introduce new concepts to the student (or review old ones). Performances are designed to either test a student's ability, serve as a drill or both. If a student passes all performances, then the student passes the lesson and the Student Model is updated based on the lesson's objectives.

3.4 The Lesson Planner

[4]Normally, one presentation followed by one performance.

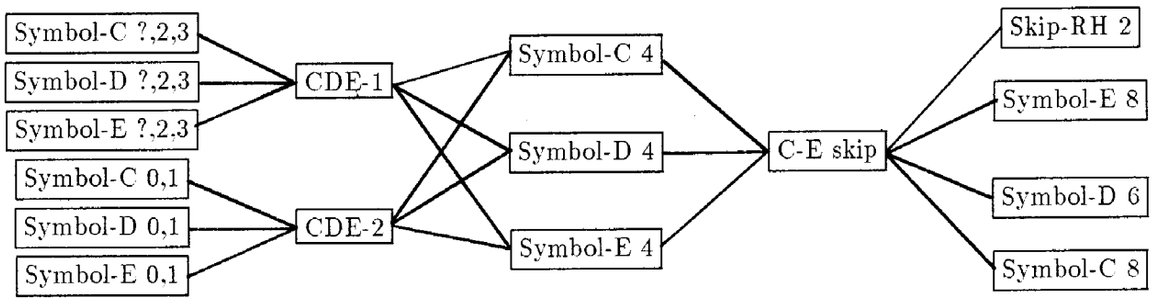

Figure 2: The Skill/Lesson graph

The Student Model and the Lesson Library form a directed graph. Figure 2 is an example of a skill/lesson graph. Lessons CDE-1 and CDE-2 have the same objectives (boosting the student's skill in playing notes C, D and E to 4), but lesson CDE-2 assumes the student has no knowledge about the piano keyboard or reading music. Lesson CDE-1 assumes the student has a little knowledge about both. In practice, a new student would be given lesson CDE-1. If he or she failed, then the proficiencies in the Student Database for the skills Symbol-C, Symbol-D and Symbol-E would be set to 0 during remediation (from their initial value of "?") – making the student eligible for lesson CDE-2. Lesson C-E skip teaches more about playing C, D and E, and introduces a new skill – skipping notes using the right hand.

It is possible to select lessons simply on the basis of analyzing the lesson/skill graph (i.e. find any lesson which has all of its prerequisites satisfied, and teaches at least one new skill). This approach, while being easy to implement, has a disadvantage: it is not guaranteed to present the lessons in a "coherent" order. Depending on the situation, it may be appropriate to concentrate on teaching one skill or to switch between teaching several different skills.

The Piano Tutor uses a rule-based system to select the next skill to teach. A lesson is then found to teach that skill. For the small number of lessons currently in the Piano Tutor, this approach seems to provide the required flexibility. In addition, selecting a skill before selecting the lesson has several advantages over selecting the lesson directly:

1. We have found that it is easier to define rules for selecting skills than it is to define rules for selecting lessons.
2. The Lesson Library may be extended without changing the skill selection rules.
3. It may be decided, through error remediation, that a particular skill needs to be taught. In these cases, the Piano Tutor skips the skill selection step and proceeds to lesson selection immediately.

Selecting the next set of skills to teach is done by using a single set of rules which can reference the Student Database. The rules themselves form a list of "if-then" statements, where the "if" clause is a LISP expression and the "then" clause is a skill to teach next. For example, a simple rule might be:

```
if (= (skill rhythm-accuracy) 6)
then (rhythm-accuracy)
```

This rule is interpreted as follows. Whenever the student's estimated proficiency at the skill rhythm-accuracy is equal to 6, teach rhythm-accuracy. Rules are evaluated sequentially, the skill selected by the first satisfied rule is returned.

Given a skill to teach, the current version of the Lesson Planner finds the first lesson in the Lesson Library that has all of its prerequisites satisfied and teaches the skill. This method of selecting the lesson was never viewed as totally satisfactory, but it did give us valuable experience for designing the next version of the Lesson Planner (see Section 6).

3.5 The Lesson Driver

The Lesson Driver:

- selects, based on the rules stored in the lesson, which set of presentations and performances are given to the student.
- gives each presentation and performance to the student.
- updates the student model, based on the lesson objectives (if all performances are passed).

Select The rules to select which introduction/performance to the student will normally be fairly simple – their primary purpose is to prevent the Piano Tutor from repeating material word-for-word. Another use for these rules, however, is that they can be used to adjust the lesson to known attributes of the student. For example, if the Student History indicates that the student has difficulty playing 16^{th} notes, then a performance could be selected which does not contain 16^{th} notes (unless, of course, the purpose of the lesson is to teach 16^{th} notes).

Present Currently, presentations in the Piano Tutor consist of ordinary text. In the future, it will be possible to mix text, graphics, sound and interactive video-disk presentations together. The Presentation Driver is responsible for taking a a list of text files, images, track numbers, etc., and showing them, in sequence, to the student.

Perform A performance in the Piano Tutor consists of the task description (the student's musical score and the accompaniment score) and remediation rules. The Performance Driver passes the task to the Performance Analyzer and then provides remediation based on the returned error log and the rules stored with the performance. If the student makes no errors, then the student passes the lesson. If the student makes some errors, then the student may pass, depending on the remediation rules.

Update If the student passes every performance in the lesson, then the Student Model is updated based on the lesson's objectives. If a student fails a performance, then the lesson ends and the update step is skipped (although the Student Model may still be changed as a result of remediation).

4 Remediation

Each performance has, in addition to the musical score which is passed to the Performance Analyzer, a set of rules which are used to analyze the student's errors. The rules have the same format as the rules used to select skills (see Section 3.4), where the action clause is the name of an error handler. We found that many of the rules were very simple (e.g. if a pitch error occurs on the 5^{th} note, then

the error is a skip error). These rules are written using a second, simplified, rule format: a list of error handlers to be called if the student makes a pitch or rhythm (starting time, duration, tempo and velocity errors are all classified as rhythm errors) error. Normally, the second rule format is all that is needed.

If the student makes more than one error, two heuristics are used to select the most "significant" error:

- Pitch errors are more significant than rhythm errors.
- The first error within an error type (pitch vs. rhythm) in more significant than any subsequent errors.

This ranking of errors is a byproduct of the way in which remediation rules are stored and evaluated. Each performance has four types of rules:

- Pitch rules
  Conventional "if-then" rules for analyzing pitch errors.
- Pitch list
  A list of error handlers for errors which are diagnosed as a pitch error on a single note.
- Rhythm rules
- Rhythm list
  The equivalents of the pitch rules and list for rhythm errors.

These four sets of rules are evaluated in sequence (first the pitch rules, then the pitch list, etc.) until an error handler is called. The error handler remediates the error and can either continue remediation (i.e. check for other errors) or stop it.

4.1 Error handling routines

Remediation in the Piano Tutor is actually performed by the error handling routines. Each error handler corresponds to a basic error type (e.g. playing an incorrect pitch when the previous pitch is not adjacent to the target pitch). Error handlers may either be global (usable by any lesson) or local (usable by only a single lesson).

In general the error handlers will present a graduated series of responses. For example, the first time the student makes an error, the error handler may simply give a warning and have the student repeat the performance. The second time, the Student History might be checked and, if the student has a history of this type of error, the student might be asked to perform a remedial lesson. The third time, the appropriate skill might be added to the skill list to guide the selection of the next lesson and the lesson ended (failing the student).

Using a graduated response seems to reduce the need to differentiate between cognitive and performance errors. If an error is a cognitive error, then it will probably be repeated, triggering more detailed remediation. If an error is a performance error, then it probably will not be repeated and the extra practice does not hurt.

Remediation takes the form of some combination of the following options:

- Have the student repeat some or all of the performance.
- Have the student execute a different lesson, and then return to the current one.

the error is a skip error). These rules are written using a second, simplified, rule format: a list of error handlers to be called if the student makes a pitch or rhythm (starting time, duration, tempo and velocity errors are all classified as rhythm errors) error. Normally, the second rule format is all that is needed.

If the student makes more than one error, two heuristics are used to select the most "significant" error:

- Pitch errors are more significant than rhythm errors.
- The first error within an error type (pitch vs. rhythm) in more significant than any subsequent errors.

This ranking of errors is a byproduct of the way in which remediation rules are stored and evaluated. Each performance has four types of rules:

- Pitch rules
  Conventional "if-then" rules for analyzing pitch errors.
- Pitch list
  A list of error handlers for errors which are diagnosed as a pitch error on a single note.
- Rhythm rules
- Rhythm list
  The equivalents of the pitch rules and list for rhythm errors.

These four sets of rules are evaluated in sequence (first the pitch rules, then the pitch list, etc.) until an error handler is called. The error handler remediates the error and can either continue remediation (i.e. check for other errors) or stop it.

4.1 Error handling routines

Remediation in the Piano Tutor is actually performed by the error handling routines. Each error handler corresponds to a basic error type (e.g. playing an incorrect pitch when the previous pitch is not adjacent to the target pitch). Error handlers may either be global (usable by any lesson) or local (usable by only a single lesson).

In general the error handlers will present a graduated series of responses. For example, the first time the student makes an error, the error handler may simply give a warning and have the student repeat the performance. The second time, the Student History might be checked and, if the student has a history of this type of error, the student might be asked to perform a remedial lesson. The third time, the appropriate skill might be added to the skill list to guide the selection of the next lesson and the lesson ended (failing the student).

Using a graduated response seems to reduce the need to differentiate between cognitive and performance errors. If an error is a cognitive error, then it will probably be repeated, triggering more detailed remediation. If an error is a performance error, then it probably will not be repeated and the extra practice does not hurt.

Remediation takes the form of some combination of the following options:

- Have the student repeat some or all of the performance.
- Have the student execute a different lesson, and then return to the current one.

- Add information to the Student History.
- Change a estimated proficiency for a skill in the Student Model.
- Present additional material to the student using the Presentation Driver.
- Replay some or all of the student's previous performance.
- Play some or all of the correct performance.
- Enqueue a skill to be taught by the next lesson.
- Continue to check for additional errors. If there are no more errors, pass the student.
- Fail the student, ending the lesson.

5 Status

The Piano Tutor is in the very early stages of development. At this time, we have implemented the Lesson Planner and Lesson Driver, as well as about a half-dozen lessons and the associated error handlers. The system is capable of presenting music notation on a computer display for a performance by a student. Errors are analyzed as described in Section 4, resulting in ordinary text comments as well as graphical highlights indicating where the error(s) occurred. Our prototype performs remediation and lesson selection based on student errors and the information maintained in the Student Model. Because of the limited repertoire of lessons, we have not yet made any trials with a "real" student.

We have also implemented a prototype Piano Practicer system, which enables a student to select a song and perform it. The Piano Practicer implements the passive mode mentioned in Section 1.2. All actions are initiated by the student, who can ask to hear the computer perform the selected piece or perform the the piece him or herself. After giving a performance, the student can ask to hear a recording of his or her most recent performance and the student can ask the computer to point out any errors that were made. Finally, the student can ask the computer to play an accompaniment.

6 Future plans

The Lesson Planner and Remediation handler are both being changed. The Lesson Planner is not flexible enough to do a "good" job of selecting lessons. In particular, it is difficult to write rules to pick the most appropriate lesson (based on information stored in the Student History) of two or more similar lessons (lessons which have the same prerequisites and objectives). The Remediation Handler does not handle "special" cases very well. For example, there are cases where it is important to distinguish between different types of rhythm errors (velocity and duration errors, for example). While this could be done using the rhythm rules, it meant that all of the rules stored in the rhythm list would have to be moved to the rhythm rules if rhythm errors were to be handled in sequence (i.e. so that an error on note 3 would be handled before an error on note 7).

The Lesson Planner is being changed as follows. First, the rule processor is being changed to incorporate iteration over sets of items. This will allow us to directly express rules like "for each eligible lesson, if the lesson teaches a skill in the skill list, then ...." Second, the rules themselves will be changed so that, rather than selecting skills and then lessons, the rules "vote" for lessons.

Once all of the rules have fired, the lesson with the most votes wins. The old Lesson Planner can be implemented using this system, and the additional flexibility will make it easier to write rules for selecting between similar lessons.

In the next version of the Piano Tutor, the format for the remediation rules will change. Rather than having four separate types of rules for each performance (see Section 4), there will be a list of error handlers for each note. Remediation will be done by making multiple passes through the notes of a student's performance. The notes are examined in sequence. On the n^{th} pass through the performance, if there are any errors on a note, the n^{th} error handler for that note is called. The error handler checks to see if the student's error is one that it should be concerned with. If not, then it returns and remediation continues. We envision that there will normally be only two passes. The first will look for pitch errors and the second will look for rhythm errors. For the sake of generality, this search for error handlers is optionally both preceded and followed by a set of rules to check for special cases.

7 Conclusions

The Piano Tutor is interesting for several reasons. First, it is a highly interdisciplinary project involving musicians, computer scientists and an instructional designer. From a technological perspective, we are combining computer controlled synthesized sound, computer graphics, videodisk, speech and a piano keyboard in an interactive system.

Within the system, we intend to support two modes of interaction. In the passive or practice mode, tasks are selected by the student and evaluations are made only on request. In the active or tutor mode, the computer selects tasks and gives feedback to the student after each performance. This approach seems to be necessary to provide balanced and enjoyable instruction.

The use of a real-time Performance Analyzer in conjunction with rule-based lesson planning and remediation seems to be an effective solution to the problem of real-time control in an intelligent system. With this approach, we obtain fast response during a real-time performance without sacrificing the use of rule-based programming for complex decision making.

We believe the Piano Tutor will be an effective new approach to piano instruction. Our hope is that we make a student's practice time more productive and also free the piano teacher to set aside mundane drills and, instead, teach the art of music.

References

[1] Joshua J. Bloch and Roger B. Dannenberg (1985) Real-time Computer Accompaniment of Keyboard Performances. *Proceedings of the International Computer Music Conference.* pp. 279-289. Computer Music Association, San Francisco, CA 94101.

[2] John R. Anderson, C. Franklin Boyle, Brian J. Reiser (1985) Intelligent Tutoring Systems. *Science.* 228, pp. 456-462.

[3] William J. Clancey (1982) Tutoring Rules for Guiding a Case Method Dialog. In D. Sleeman and J.S. Brown (Editors) *Intelligent Tutoring Systems.* pp. 201-225. Academic Press: New York.

[4] David Hon (1982) Interactive Training in Cardiopulmonary Resuscitation *Byte* vol 7, number 6. pp. 108-138. Academic Press: New York.

[5] George Orwell (1948) *1984.* Harcourt Brace Jovanovich: San Diego.

[6] Ira P. Goldstein(1982) The genetic graph: a representation for the evolution of procedural knowledge. In D. Sleeman and J.S. Brown (Editors) *Intelligent Tutoring Systems.* pp. 51-77. Academic Press: New York.

[7] Jesse M. Heines and Tim O'Shea (1985) The design of a rule-based CAI tutorial. *International Journal of Man-Machine Studies.* 23, pp. 1-25.

Reference:
Dannenberg, R. B., M. Sanchez, A. Joseph, R. Joseph, R. Saul, and P. Capell. Results from the Piano Tutor Symposium, D. Smalley, N. Zahler, and P. Galvani, eds. New London, Conn.: Connecticut College, 1993. pp. 143-150