Incorporating Music into the Study of Algorithms and Computer Programming

Takeo Tatsumi
Tokyo University of Agriculture and Technology
2-24-16 Naka-Cho, Koganei, Tokyo 184-8588, Japan
takeo@tt.tuat.ac.jp

Yoshiaki Nakano
Senri-Kinran University
5-25-1 Fujishirodai, Suita, Osaka 565-0873, Japan
y-nakano@kinran.ac.jp

Kiyoshi Tajitsu
Hokusei-Gakuen University
2-3-1 Ohyachi-Nishi, Atsubetsu, Sapporo, Hokkaido 004-8631, Japan
tajitsu@hokusei.ac.jp

Haruhiko Okumura
Mie University
1577 Kurimamachiyao, Tsu, Mie, 514-8507 Japan
okumura@edu.mie-u.ac.jp

Yasunari Harada
Waseda University
1-6-1 Nishi-Waseda, Shinju-ku, Tokyo 169-8050, Japan
harada@waseda.jp

ABSTRACT

According to the national syllabus set forth by the Ministry of Education, Culture, Sports, Science and Technology, or hence force MEXT for short, and implemented in senior high schools since the school year of 2003-2004 in Japan, every student is expected to study and acquire necessary credits for the newly established subject Information Study. Learning about algorithms, with some exposure to and experience with programming languages, is part and parcel of the 'Scientific Aspect,' one of the three major goals set forth by MEXT, of this new subject. When learning a computer programming language, students must get accustomed to many new concepts, terms, and definitions along with the syntax and semantics of this language. There are too many elements to comprehend and master at the same time and this is part of the reasons why the number of students who study algorithms in K-12 is not increasing despite the introduction of Information Study in Japan. Noticing some apparent similarities between learning musical scores and learning computer programming languages, the authors proposed that incorporating study of music is a reasonable and interesting new way in learning programming languages and algorithms for senior high school students. In this paper, we describe our field trial based on this idea and suggest reasons why we believe this musical approach might be effective in learning the 'Scientific Aspect' of Information Study.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education

General Terms

Compiler / Programming Languages & Paradigms

Keywords

K-12 Instruction

ICMI-MLMI '09 Workshop on Child, Computer and Interaction November 5, 2009, Cambridge, MA, USA

1. INTRODUCTION

In this section, we offer a general overview of ICT education in Japan in order to place our proposal in a proper context and perspective.

1.1 ICT and Education

In recent years, the development of information and communication technology (ICT) is changing our daily lives and our ways of thinking. Education is affected as are many other aspects of our lives. Teachers and administrators seek efficiency in utilizing ICT for education but the impact of these societal changes to teaching and learning caused by the rapid development of ICT should be considered at least in two distinct aspects. On the one hand, schools and educational institutions must introduce ICT in their teaching along with their organizational administration to modernize daily activities and operation in schools. On the other hand, they have to teach their students how to deal with ICT. Because of some unfortunately wordings in Japanese relating to these discussions, the Japanese expression referring to "information education" is somewhat ambiguous between teaching of ICT and teaching with ICT, and many researchers in the field are conflating those two concepts. Although "education utilizing ICT" is important for obvious reasons, we believe that "ICT education" in the sense of "teaching of ICT" is no less important.

1.2 ICT Education in Japan

In 1998, Japanese Ministry of Education, Culture, Sports, Science and Technology published a revised National Syllabus for K-12 education to be fully implemented in elementary schools and junior high schools starting in the school year of 2002-2003 and in senior high schools in the school year of 2003-2004. There were a number of innovative features in this revision, some of which still remain controversial.

First, usage and utilization of ICT in all subject areas were emphasized to augment familiarity with and literacy in computer and networks among teachers and students. Use of computers and networks in existing subjects such as language, math, science, social study, arts, music and physical education were encouraged with additional funding for equipment and facility such as personal computers, network resources, and educational and administrative software. Second, classes for "comprehensive study" was introduced from elementary to senior high schools, in which students would engage in research, presentation and summary of their research results in such cross-subject topics as environmental study and international awareness to supplement traditional subjects such as language, math, science and social study. Naturally, use of the Internet and personal computers were encouraged for research and summary and use of presentation tools were encouraged for classroom discussions. Third, a completely new subject of Information Study was introduced in senior high schools.

There were three goals for this new subject of Information Study set forth in this revision. First is the augmented literacy in computer usage and network communications. Second is understanding of the 'Scientific Aspect' of Information. Third is cultivating readiness for participating in the global information society today.

Although "understanding of basic algorithms" is specified as an obligatory item in the understanding of the Scientific Aspect of Information Study in the revised National Syllabus, most teachers in charge of this subject are not ready to teach such topic area, while they are willing to teach intricate skills for using Office applications because they generally think that use of those proprietary software products are indispensable and important for senior high school students when they enter the real world.

1.3 Education of Computer Programming Languages

Most teachers at K-12 in Japan do not even consider teaching computer programming languages to their students in class because they are not familiar with programming or algorithms themselves. When some do teach programming in class, they tend to set their goals too high, with too much expectations for the students and using textbooks that are too difficult for beginners. In most cases, classes for computer programming are tedious because teachers in charge of those classes, more often than not, lack experience or expertise in getting their students involved or having their students enjoy what they are doing, as compared to teachers in art, music and sports.

In our view, this is a serious drawback of the current realities of ICT education in Japan because writing a working piece of computer program is essential in understanding algorithms and how computers work and realizing how people make those automated machineries function as expected.

There are a number of possible approaches for teaching a complete novice how to write a piece of working code in a programming language. One possible way is to show and give a functioning program and let the students change various parameters or pieces and let them see how those changes affect the outcome. In this approach, one can take a relatively complex code from the beginning and let the students work on particular parts and have them understand particular details of the programming language in question. Another possible approach would be to explain how a piece of algorithm should work in flow charts or some other manner and explain how that should be translated in a specific language. You might want to explain different programming languages at the same time and emphasize that algorithms are the essential part of a program and they can be implemented in various different ways. We believe that letting the students write a piece of working code in a programming language from scratch is actually going to be the most effective approach for novice learners of programming languages. Although students may be puzzled and misled by various minor details, they would learn to pay attention to the smallest details of a program such as commas, periods and other delimiters. The resulting pieces of code in this approach should naturally be small, but the students' understanding would be deep.

In order for this to happen, the programming language to be used for this educational purpose must be simple in syntax, without too much complication.

This is similar to teaching students how to write a web page. A quick and easy way might be to use some word processing software and save the document as a web document but the students would have little understanding of how web browsers work. Writing a web page from scratch using some simple text editors, without using any web editors, would be slightly more difficult for the students and the resulting pages may be simpler, but the students will more fully understand how HTML tags work.

2. MUSICAL METHOD IN ICT EDUCATION

In this section, we suggest how understanding of musical scores might help students learn computer programming languages and algorithms.

2.1 Music education utilizing ICT

In recent years, music education is changing through use of computers in classrooms. Before this change, both teachers and students had to use pencil or pen to write musical scores. With personal computers and score editing software, music teachers now can more easily write down musical notes for their students just as language teachers can use word-processing software and math teachers can use graphics software in preparing teaching materials. In an article on the BBC's web page[1], we find a description of a music teacher in England[1].

> Computers are playing an increasing role in music, especially in secondary schools. Now software can enable students to compose and arrange music and develop their musical creativity, even when they can't play an instrument. But provision is patchy and can depend upon the enthusiasm of your child's music teacher.
>
> - There were clear benefits for student motivation and attainment
> - When used appropriately ICT can deepen pupils' understanding of music without be-

[1] http://www.bbc.co.uk/music/parents/features/computers_school.shtml

ing reliant on the pupil's physical ability to perform the music written

- Students made the best use of ICT when they had been taught the building blocks needed to compose. Musical software alone was not enough to ensure success

Thus, ICT can play a very important role in music education. However, the goal of music education utilizing ICT is different from what we propose below, namely utilization of music in the education of ICT.

There are many structural notations in musical scores, which teachers fluent in programming languages can readily understand. In 1998, Tatsumi argued positive potentials of musical approach to computer programming education[5], noting that musical scores are written using specific rules: iteration, jump, skipping and other controls, all of which are also used in computer programs. Understanding these rules in musical scores can help students in reading a computer program. This is why musical method could be a good candidate for developing a new approach for learning computer programming. In October 2001, Tatsumi argued for a "musical method in ICT Education" [4], but the actual classroom implementation of the idea came later.

2.2 Desiderata for Computer Programming Education

Many teachers in charge of ICT education in Japan erroneously regard computer programming education in K-12 as a kind of vocational training. The number of students who will go on to be professional programmers is limited while the rest of the students will get jobs outside the core of the ICT industry where their jobs will not be directly connected to writing programs or designing algorithms. Actually, we believe that learning about algorithms and how to write a piece of working code is important for all senior high students, regardless of the kinds of jobs they will get in the real world, for better understanding how automated systems work and how people make them function as they do.

Hands-on experiences are important in learning any subject matter, either in schools or outside,, as in learning how to cook, learning how to conduct chemistry experiments and learning how to do math and algebra. Therefore, we think the experience of writing a piece of computer program is very important for all K-12 students. For instance, such an experience helps students to find the source of a computer trouble.

Here, we list important considerations in preparing a successful computer programming course.

Let your students experience important notions in computer programming.

To write a program from scratch, students have to learn such things as models of data structure, data flow, sequential orders, iteration, exiting loop, "if-then-else" clause, functions, recursions and synchronizations. We think that two or three notions above should be enough for a K-12 course for students who are new to programming.

Let your students experience an entire programming project cycle, however small, consisting of writing, testing and debugging.

Most students do not know the simple fact of life that human programmers write computer programs. They think that computer programs are generated through some automatic procedures. We believe that going through a project cycle of writing a piece of code from scratch, testing and debugging the code and then rewriting, testing and debugging the revised versions of the code is effective in getting a sense of how it is that computers function the way they do.

Use a system that corrects minor syntax errors automatically.

In order to let all kinds of k-12 students experience such a project cycle, utilizing a few of the key concepts from among data structure, iteration, synchronization and so on, we need a programming language with simple and easy-to-understand syntax. In some programming languages, students must write many lines for a simple algorithm. It is going to be hard and tedious work for beginners when they have to deal with minor spelling or syntax errors in writing a program in such languages. For example, beginners may type a semi-colon when they must type a colon, or vice versa. This hinders students from concentrating on the semantics of their programs. At the initial stages of learning programming, students must get used to using editors, memorize the grammar of the programming language and understand the semantics at the same time and this could be a complex task for K-12 students. A compiler or an interpreter that automatically corrects small syntax errors should be desirable for these purposes.

Reduce your students' fears that they might physically break down computers or some equipment through some mistakes.

When writing a piece of code that controls physical devices, students are inherently afraid that their program might break down those devices. It would give them additional sense of security if they knew that their programs would not possibly break down any physical devices. For example, turtle graphics or playing music are better candidates than manipulation of cars, soccer robots or helicopters for students who are new to programming. When they make a semantic error in musical programming, the worst it can causes would be a strange series of discordant sounds, or lack thereof, and they can feel relatively assured that no devices are going to break down. There is no need for students to hesitate testing their programs to find semantics errors on graphics or music.

3. PROGRAMMING LANGUAGE "DOLITTLE"

Dolittle, developed by K. Kanemune and Y. Kuno[3, 2], is an object oriented programming language intended for beginners in programming that meets the four desiderata mentioned above. Dolittle, which refers both to the language and to an interactive environment that interprets it, was originally designed with the intention that students can learn how to handle turtle graphics easily when they use this language/system. In summer of 2005, Kanemune and Kuno added several new functions for playing simple musical notes. In September 2005, Tatsumi proposed additional

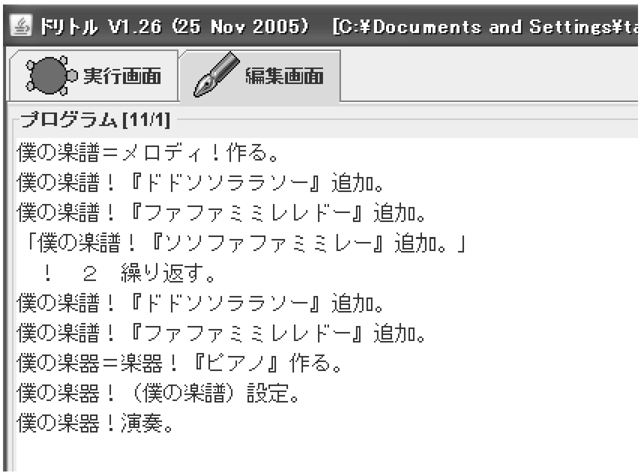

Figure 1: Screen Shot of Dolittle

musical functions for Dolittle for handling complex musical scores. Today, we can write Dolittle programs for playing musical scores with rich control structures.

3.1 Melody Objects

In Dolittle, [2] objects are defined on the basis of prototype objects, which are a kind of 'reserved objects.' As an example, we can define the following melody object [3] which is generated on the basis of several prototype objects.

```
Twinkle = melody ! create.
Twinkle ! "C C G G A A G - F F E E D D C -" add.
Twinkle ! "G G F F E E D - G G F F E E D -" add.
Twinkle ! "C C G G A A G - F F E E D D C -" add.
MyInstrument = instrument ! "Piano" add.
MyInstrument ! (Twinkle) set.
MyInstrument ! play.
```

3.2 Iteration in Music

The fifth line can be represented as follows using iteration, yielding a more structured version of the same song.

```
Melody-X = "C C G G A A G - F F E E D D C -".
Melody-Y = "G G F F E E D -".
Twinkle = melody ! create.
Twinkle ! (Melody-X) add.
{Twinkle ! (Melody-Y) add.} ! 2 repeat.
Twinkle ! (Melody-X) add.
MyInstrument = instrument ! "Piano" add.
MyInstrument ! (Twinkle) set.
MyInstrument ! play.
```

3.3 If-Then-Else Clause

We can write a slightly more "structured" code for the same song, using if-then-else and an incremental counter. [4]

```
Melody-X = "C C G G A A G - F F E E D D C -".
Melody-Y = "G G F F E E D -".
Twinkle = melody ! add.
{ |counter|
  Twinkle ! (Melody-X) add.
  if "counter == 1" ! then {
    {Twinkle ! (Melody-Y) add.} ! 2 repeat.
  } done.
} ! 2 repeat.
MyInstrument = instrument ! "Piano" add.
MyInstrument ! (Twinkle) set.
MyInstrument ! play.
```

3.4 Abstraction of Structure from a Song

Twinkle Star has a structure of the form X-Y-Y-X, where X is "C C G G A A G - F F E E D D C -" and Y is "G G F F E E D -". "Summ Summ Summ", which was written by Hoffmann von Fallersleben, an old German poet, is very famous in Japan. "Summ Summ Summ" also has a structure of the form X-Y-Y-X and can be played only by replacing Melody-X and Melody-Y. Students can thus understand the abstract nature of the structure of a song.

```
Melody-X = "G G F F E E R R D E F D C C R R".
Melody-Y = "E F G E D E F D".
Twinkle = melody ! add.
{ |counter|
 Twinkle ! (Melody-X) add.
 if "counter == 1" ! then {
   {Twinkle ! (Melody-Y) add.} ! 2 repeat.
 } done.
} ! 2 repeat.
MyInstrument = instrument ! "Piano" add.
MyInstrument ! (Twinkle) set.
MyInstrument ! play.
```

[2] Dolittle is a Japanese programming language. The codes actually written are represented in (Chinese and) Japanese characters. The field trial classes were conducted in Japanese and the program pieces are written in Japanese. The example codes are translations of those codes.

[3] Actually, "melodies" in the following should more correctly be termed as "phrases". As not many of the students who participated in the field trial discussed in section 4 have learned theory of music, we used a more familiar expression for the general students.

[4] In a different field trial conducted in the following year, we changed the sequence for those melodies to produce different songs.

3.5 Random Number: Automatic Composition

The following is an example of automatic composition in Okinawan scale using random numbers and an array. Okinawa is the southernmost area of Japan, with its indigenous language, culture and musical scale. The Okinawan scale is pentatonic, with only five notes, namely C, E, F, G, and B, in each octave.

```
Okinawan = array ! create.
Okinawan ! 'C' inserted 'E' inserted 'F' inserted
  'G' inserted 'B' inserted .
MyScore = melody ! create.

Joyful = melody ! create.
{joyful ! (Okinawan! (random(5)) lookup) add.}
   ! 6 repeat .
Joyful! 'C -' added .
Sad = melody ! create.
{sad ! (Okinawan! (random(5)) lookup) add.}
   ! 8 repeat .

MyInstrument = instrument ! 'GUITAR' create.
{MyScore ! (Joyful) add.} ! 2 repeat .
MyScore ! (Sad) add.
MyScore ! (Joyful) add.

MyInstrument ! (MyScore) set.
MyInstrument ! play.
```

3.6 Ensemble: Concurrency in Programs

Dolittle: you can write concurrency.

When you write a program for ensemble, you create many musical instrument objects and one band object.

```
PAT-score = melody ! create.
{
 PAT-score ! "C C G G A A G -" add.
 PAT-score ! "F F E E D D C -" add.
} ! 2 repeat.
PAT-Instrument = Instrument ! "Piano" create.
PAT-Instrument ! (PAT-score) set.
```

```
ERIC-score =  melody ! create.
{
 ERIC-score ! "C D G A A F G -" add.
 ERIC-score ! "F G G A A G G -" add.
} ! 2 repeat.
ERIC-Instrument = Instrument ! 50 create.
ERIC-Instrument ! (ERIC-score) set.
```

```
MyBand = band ! create.
MyBand ! (PAT-Instrument) add.
MyBand ! (ERIC-Instrument) add.
MyBand ! 108 tempo.
MyBand ! play.
```

Students can play with the program and interactively see how a change in their code is reflected in the resulting songs.

4. FIELD TRIALS IN TWO JAPANESE SENIOR HIGH SCHOOLS

In this section, we report our experiences in the two field trials of utilizing this musical approach for teaching ICT using Dolittle in Japanese senior high schools.

4.1 Our Goals and plans.

We prepared textbooks and other source materials before the field trial in the first high school, then revised them for the trial in the second high school and evaluated their usability and appropriateness.

- Activities in two different kinds of senior high schools.
- Teachers' expertise in music is critical. In these field trials, the two high school teachers involved, Mr. Ohara and Ms. Nobe, had enough background knowledge for teaching in their classes.
- We wrote a textbook of ICT incorporating our musical approach. Through these trials, we continued to revise the textbook.
- We set the goals for the students in our field trials as follows:
  - Offering hands-on experiences in computer programming
  - Introducing computer programming
  - Cultivating good attitude toward copyright issues
  - Enriching appreciation of music

4.2 Environments of the field trials

There are many ideas / devices / toys proposed for education of computer programming. For example, Lego Mindstorm is a programmable robotics kit. There are many other educational toys of this kind. To use these toys in a class, the school must provide room large enough to let the robots run around and secure substantial budget to purchase those toys. Before the class, teachers must spend many hours making all kinds of preparations such as drawing course lines for the robots to follow. On the other hand, our approach needs only the following:

- PC which can run Java with audio devices. The programming environment "Dolittle" runs on two platforms.
- Java JRE on Windows/Mac OS X/Linux
- Java Applet on web browser.
- "Dolittle" is free software.
- loudspeaker which is connected to the teacher's PC.

Figure 2: Machida High School

Figure 3: Momodani High School

- headphones /earphones for students. In Japanese high schools, headphones are standard equipment in all PC rooms used for language learning, so it is not difficult to secure such classrooms for the trial.

We do not have to buy any additional devices and/or toys. We do not need any floor space for the robots to run around. In comparison with other approaches, the additional cost of our trials is relatively low. Even if a school has small budget, a teacher with enough enthusiasm can try this experience.

4.3 Machida High School

Machida High School is located in Tokyo Prefecture. Almost 100 percent of the students who graduate from this school go on to enter universities. Mr. Tsutomu Ohara, who teachers Information Study in this school, had written many musical computer programs while he was a student. The field trial in this school was set as a supplementary lecture with optional attendance in June of 2007. As it turned out, more students wanted to participate than we had anticipated. There were 29 female and 13 male students present in the class. As there are more male students in this school than female students, this shows that a greater portion of female students decided to attend this class than male students. We had told the students that those who were to attend the class had better be able to read musical scores. There is some gender disparity regarding literacy in music scores and this seems to have affected the attendance to this field trial.

In this class, students used an online-version of Dolittle[5] which runs within web browser with Java JRE. There is no need to install Dolittle onto the PCs of the classroom. However we must provide tips on copying and pasting the programs.

- 1st day (70 min): introduction to Dolittle.
- 2nd day (70 min): registration to CMS(Moodle), iteration of notes in the score, "if-then-else"
- 3rd day(100 min): transcribing their favorite songs in Dolittle.

We didn't write sample scores of ensemble in the first version of the textbook so when those students wanted to play ensemble, they wanted to see some sample codes but there weren't any. Lack of those sample codes seems to have inconvenienced them.

4.4 Momodani High School

Momodani High School is located in downtown Osaka City. Only a small fraction of its graduates go on to a University, which suggests that teachers have to work hard to motivate their students in learning something new. Ms. Midori Nobe teaches Information Study in this school. Her trial class was set within a regular course in October 2007. There were about 20 students in her class, all of whom had to attend the trial class.

In her class, students used the stand-alone version of Dolittle. Students can save and load programs from / onto files easily. However, the teacher had to install Dolittle beforehand. We used a revised version of the textbook.

- How to save and load their program.
- Add sample ensemble programs

There were many students who were not particularly interested in music as this trial was embedded in a regular course. So, Ms. Nobe prepared and provided several worksheets for her students. Ms. Nobe wrote several exercises in these worksheets. These exercises were related to the comprehension of ICT.

- We instructed the students to draw flow-charts using musical scores.
- We asked the students to consider how they would represent different lengths of various musical notes in Dolittle.

These worksheets were helpful for the students in making their programs.

[5] `http://dolittle.eplang.jp/pconline/applet.html`

Figure 4: Helpful WORK SHEET

Figure 5: Helpful WORK SHEET

4.5 Results

We summarize the results of these experimental classes on this approach as we found them from written responses from the students who participated in those trial classes and from observations by the teachers in charge.

4.5.1 *Changes of Students' Attitude*

Strongly encouraging: Students in both high schools constructed programs eagerly. Attitudes of Momodani High School students changed noticeably in this approach. Some Momodani High School teachers were deeply surprised at the change of their students' attitudes.

4.5.2 *Copyright Issue*

Strongly encouraging: Many Japanese high school students download unauthorized duplication of MP3 files of popular music. Ms. Nobe had taught earlier in her class that the copyright issue is important for music. After they understood the difficulties in composing original music, we found that some students reconsidered seriously about the MP3 files in their music players. We can safely say that some students changed their attitudes toward the copyright issues of music because they wrote programs for playing music.

4.5.3 *Transcription by hands*

Strongly encouraging: Through this practice, most of the students in our classes understood that someone have to transcribe musical scores at some point in order for the music to be played by a computer. This made them realize that computer programs cannot be generated automatically. We consider this realization very important.

4.5.4 *Acquisition of Musical Ability*

Strongly encouraging: Before this class, some Momodani High School students were not able to read musical scores. Mr. Nobe's worksheet was helpful for them. After this class, they were able to read musical scores. We think that the experience in this class helped them develop this ability.

4.5.5 *Gender-specific preference*

Somewhat unexpected: In Japan, many female students tend to avoid studying mathematics, science and computer technology seriously. Generally speaking, male students enjoy programming exercises while female students do not. In Machida High School, many female students were charmed with musical programming. Mr. Ohara remarked that it was the first experience in his teacher life to see female students enjoy programming. We suspect that one reason might be that many of the female students had learned playing piano in their primary school ages. If such gender differences are a part of challenges for Japanese school education, musical approach to computer programming education may help reduce one area of gender disparity by integrating two subject areas, programming where female students seem to have "social hindrance" and music training where female students seem to have some "social edge".

4.5.6 *Dependent on students' musical abilities*

Not encouraging but as expected: In Momodani High School, several students did not like music at all. These students were not interested in this class. After this class, they remarked that they did not enjoy that experience. Before this trial, we had expected such students. We cannot avoid this problem but we think that other approaches would also have similar problems. For example, a student who is not interested in drawing cannot enjoy Turtle-graphics programming as in Logo. Students who are not interested in robotics cannot enjoy Lego Mindstorms. We think that every approach would have some such problems. We proposed just one answer for beginners of computer programming education. On the other hand, in the case of Momodani High School, as the trial was part of a regular mandatory course, some students were not interested in music. After the trial, however, some of those started to show interests in music, which is not at all surprising.

5. CONCLUSION

We proposed the musical approach for computer programming education. The premises of this approach are as follows:

1. Good integrated environments of computer programming for beginners.
2. Teachers understand that music scores and computer programs follow similar grammatical principles.

We conclude that this our initial field trials were successful in showing that:

- musical approach can be enjoyable because students find enjoyment in the study programming.
- after students were devoted to writing musical scores, they understood that computer programs in the real world are made by human efforts.
- students realized that copyright issues are important.
- the preparing cost of this approach much lower than other approaches using robotics toy, cars and so on. We do not need any additional floor space for such robots or toys.

6. ACKNOWLEDGMENTS

We would like to express our gratitude for the support offered by Prof. Susumu Kanemune and Prof. Yasushi Kuno. This project was supported in part by YAMAHA Music Foundation.

7. ADDITIONAL AUTHORS

Additional authors: Midori Nobe (Momodani High School of Osaka Prefecture, email: `y-nobe@momodani.osaka-c.ed.jp`), Tsutomu Ohara (Machida High School of Tokyo Prefecture, email: `ohara@johoka.info`).

8. REFERENCES

[1] BBC. Parents' music room - computers and music in schools, 2002.

[2] S. Kanemune and Y. Kuno. Dolittle: an object-oriented language for K12 education. In *EuroLogo2005, Warszawa, Poland*, 2005.

[3] S. Kanemune, T. Nakatani, R. Mitarai, S. Fukui, and Y. Kuno. Dolittle — experiences in teaching programming at K12 schools. In *IEEE 2nd Int. Conf. on Creating, Connecting and Collaborating through Computing*, volume C5, pages 177–184. IEEE, 2004.

[4] T. Tatsumi. Musical method in ict education(in Japanese). In *Research report on Computer and Education 2001–CE–61*, number 101 in 2001, pages 39–46. Information Processing Society Japan, 2001.

[5] T. Tatsumi and K. Kakehi. What subject of epl should be taught in high school.(in Japanese). In *Summer Programming symposium 1998*, pages 55–66. Information Processing Society Japan, 1998.