BREAKING ACCESSIBILITY BARRIERS
Computational Intelligence in Music Processing for Blind People

Wladyslaw Homenda
Faculty of Mathematics and Information Science
Warsaw University of Technology, pl. Politechniki 1, 00-661 Warsaw, Poland

Keywords: Knowledge processing, music representation, music processing, accessibility, blindness.

Abstract: A discussion on involvement of knowledge based methods in implementation of user friendly computer programs for disabled people is the goal of this paper. The paper presents a concept of a computer program that is aimed to aid blind people dealing with music and music notation. The concept is solely based on computational intelligence methods involved in implementation of the computer program. The program is build around two research fields: information acquisition and knowledge representation and processing which are still research and technology challenges. Information acquisition module is used for recognizing printed music notation and storing acquired information in computer memory. This module is a kind of the paper-to-memory data flow technology. Acquired music information stored in computer memory is then subjected to mining implicit relations between music data, to creating a space of music information and then to manipulating music information. Storing and manipulating music information is firmly based on knowledge processing methods. The program described in this paper involves techniques of pattern recognition and knowledge representation as well as contemporary programming technologies. It is designed for blind people: music teachers, students, hobbyists, musicians.

1 INTRODUCTION

In this paper we attempt to study an application of methods of computational intelligence in the real life computer program that is supposed to handle music information and to provide an access for disabled people, for blind people in our case. The term *computational intelligence*, though widely used by computer researchers, has neither a common definition nor it is uniquely understood by the academic community. However, it is not our aim to provoke a discussion on what artificial intelligence is and which methods it does embed. Instead, we rather use the term in a common sense. In this sense intuitively understood knowledge representation and processing is a main feature of it. Enormous development of computer hardware over past decades has enabled bringing computers as tools interacting with human partners in an intelligent way. This required, of course, the use of methods that firmly belong to the domain of computational intelligence and widely apply knowledge processing.

Allowing disabled people to use computer facilities is an important social aspect of software and hardware development. Disabled people are faced problems specific to their infirmities. Such problems have been considered by hardware and software producers. Most important operating systems include integrated accessibility options and technologies. For instance, Microsoft Windows includes Active Accessibility techniques, Apple MacOS has Universal Access tools, Linux brings Gnome Assistive Technology. These technologies support disabled people and, also, provide tools for programmers. They also stimulate software producers to support accessibility options in created software. Specifically, if a computer program satisfies necessary cooperation criteria with a given accessibility technology, it becomes useful for disabled people.

In the age of information revolution development of software tools for disabled people is far inadequate to necessities. The concept of music processing support with a computer program dedicated to blind people is aimed to fill in a gap between requirements and tools available. Bringing accessibility technology to blind people is usually based on computational intelligence methods such as pattern recognition and knowledge representation. Music processing com-

Homenda W. (2007).
BREAKING ACCESSIBILITY BARRIERS - Computational Intelligence in Music Processing for Blind People.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 32-39
DOI: 10.5220/0001644700320039
Copyright © SciTePress

puter program discussed in this paper, which is intended to contribute in breaking the accessibility barrier, is solely based on both fields. Pattern recognition is applied in music notation recognition. Knowledge representation and processing is used in music information storage and processing.

1.1 Notes on Accessibility for Blind People

The population of blind people is estimated to up to 20 millions. Blindness, one of most important disabilities, makes suffering people unable to use ordinary computing facilities. They need dedicated hardware and, what is even more important, dedicated software. In this Section our interest is focused on accessibility options for blind people that are available in programming environments and computer systems.

An important standard of accessibility options for disabled people is provided by IBM Corporation. This standard is common for all kinds of personal computers and operating systems. The fundamental technique, which must be applied in blind people aimed software, relies on assigning all program functions to keyboard. Blind people do not use mouse or other pointing devices, thus mouse functionality must also be assigned to keyboard. This requirement allows blind user to learn keyboard shortcuts which activates any function of the program (moreover, keyboard shortcuts often allow people with good eyesight to master software faster then in case of mouse usage). For instance, Drag and Drop, the typical mouse operation, should be available from keyboard. Of course, keyboard action perhaps will be entirely different then mouse action, but results must be the same in both cases. Concluding, well design computer program must allow launching menus and context menus, must give access to all menu options, toolbars, must allow launching dialog boxes and give access to all their elements like buttons, static and active text elements, etc. These constraints need careful design of program interface. Ordering of dialog box elements which are switched by keyboard actions is an example of such a requirement.

Another important factor is related to restrictions estimated for non disabled users. For instance, if application limits time of an action, e.g. waiting time for an answer, it should be more tolerant for blind people since they need more time to prepare and input required information.

Application's design must consider accessibility options provided by the operating system in order to avoid conflicts with standard options of the system. It also should follow standards of operating system's accessibility method. An application for blind people should provide conflict free cooperation with screen readers. It must provide easy-to-learn keyboard interface duplicating operations indicated by pointing devices.

Braille display is the basic hardware element of computer peripherals being a communicator between blind man and computer. It plays roles of usual screen, which is useless for blind people, and of control element allowing for a change of screen focus, i.e. the place of text reading. Braille display also communicates caret placement and text selection.

Braille printer is another hardware tool dedicated to blind people. Since ordinary printing is useless for blind people, Braille printer punches information on special paper sheet in form of the Braille alphabet of six-dots combinations. Punched documents play the same role for blind people as ordinary printed documents for people with good eyesight.

Screen reader is the basic software for blind people. Screen reader is the program which is run in background and which captures content of an active window or an element of a dialog box and communicates it as synthesized speech. Screen reader also keeps control over Braille display communicating information that is simultaneously spoken.

Braille editors and converters are groups of computer programs giving blind people access to computers. Braille editors allow for editing and control over documents structure and contents. Converters translate ordinary documents to Braille form and oppositely.

1.2 Notes on Software Development for Blind People

Computers become widely used by disabled people including blind people. It is very important for blind people to provide individuals with technologies of easy transfer of information from one source to another. Reading a book becomes now as easy for blind human being as for someone with good eyesight. Blind person can use a kind of scanning equipment with a speech synthesizer and, in this way, may have a book read by a computer or even displayed at a Braille display. Advances in speech processing allow for converting printed text into spoken information. On the other hand, Braille displays range from linear text display to two dimensional Braille graphic windows with a kind of gray scale imaging. Such tools allow for a kind of reading or seeing and also for editing of texts and graphic information.

Text processing technologies for blind people are now available. Text readers, though still very expensive and not perfect yet, becomes slowly a standard tool of blind beings. Optical character recognition, the heart of text readers, is now well developed technology with almost 100% recognition efficiency. This perfect technology allows for construction of well working text readers. Also, current level of development of speech synthesis technology allows for acoustic communicating of a recognized text. Having text's information communicated, it is easy to provide tools for text editing. Such editing tools usually use a standard keyboard as input device.

Text processing technologies are rather exceptions among other types of information processing for blind people. Neither more complicated document analysis, nor other types of information is easily available. Such areas as, for instance, recognition of printed music, of handwritten text and handwritten music, of geographical maps, etc. still raise challenges in theory and practice. Two main reasons make that software and equipment in such areas is not developed for blind people as intensively as for good eyesight ones. The first reason is objective - technologies such as geographical maps recognition, scanning different forms of documents, recognizing music notation are still not well developed. The second reason is more subjective and is obvious in commercial world of software publishers - investment in such areas scarcely brings profit.

2 ACQUIRING MUSIC INFORMATION

Any music processing system must be supplied with music information. Manual inputs of music symbols are the easiest and typical source of music processing systems. Such inputs could be split in two categories. One category includes inputs form - roughly speaking - computer keyboard (or similar computer peripheral). Such input is usually linked to music notation editor, so it affects computer representation of music notation. Another category is related to electronic instruments. Such input usually produce MIDI commands which are captured by a computer program and collected as MIDI file representing live performance of music.

Besides manual inputs we can distinguish inputs automatically converted to human readable music formats. The two most important inputs of automatic conversion of captured information are automatic music notation recognition which is known as Optical Music Recognition technology and audio music recognition known as Digital Music Recognition technology. In this paper we discuss basics of automatic music notation recognition as a source of input information feeding music processing computer system.

2.1 Optical Music Recognition

Printed music notation is scanned to get image files in TIFF or similar format. Then, OMR technology converts music notation to the internal format of computer system of music processing. The structure of automated notation recognition process has two distinguishable stages: location of staves and other components of music notation and recognition of music symbols. The first stage is supplemented by detecting score structure, i.e. by detecting staves, barlines and then systems and systems' structure and detecting other components of music notation like title, composer name, etc. The second stage is aimed on finding placement and classifying symbols of music notation. The step of finding placement of music notation symbols, also called segmentation, must obviously precede the step of classification of music notation symbols. However, both steps segmentation and classification often interlace: finding and classifying satellite symbols often follows classification of main symbols.

2.1.1 Staff Lines and Systems Location

Music score is a collection of staves which are printed on sheets of paper, c.f. (Homenda, 2002). Staves are containers to be filled in with music symbols. Stave(s) filled in with music symbols describe a part played by a music instrument. Thus, stave assigned to one instrument is often called a part. A part of one instrument is described by one stave (flute, violin, cello, etc.) or more staves (two staves for piano, three staves for organ).

Staff lines location is the first stage of music notation recognition. Staff lines are the most characteristic elements of music notation. They seem to be easily found on a page of music notation. However, in real images staff lines are distorted raising difficulties in automatic positioning. Scanned image of a sheet of music is often skewed, staff line thickness differs for different lines and different parts of a stave, staff lines are not equidistant and are often curved, especially in both endings of the stave, staves may have different sizes, etc., c.f. (Homenda, 1996; Homenda, 2002) and Figure 1.

Having staves on page located, the task of system detection is performed. Let us recall that the term system (at a page of music notation) is used in the meaning of all staves performed simultaneously and joined together by beginning barline. Inside and ending bar-

Figure 1: Examples of real notations subjected to recognition.

lines define system's structure. Thus, detection of systems and systems' structure relies on finding barlines.

2.1.2 Score Structure Analysis

Sometimes one stave includes parts of two instruments, e.g. simultaneous notation for flute and oboe or soprano and alto as well as tenor and bass. All staves, which include parts played simultaneously, are organized in systems. In real music scores systems are often irregular, parts which not play may be missing.

Each piece of music is split into measures which are rhythmic, (i.e. time) units defined by time signature. Measures are separated from each other by barlines.

The task of score structure analysis is to locate staves, group them into systems and then link respective parts in consecutive systems. Location of barlines depicts measures, their analysis split systems into group of parts and defines repetitions.

2.1.3 Music Symbol Recognition

Two important problems are raised by symbol recognition task: locating and classifying symbols. Due to irregular structure of music notation, the task of finding symbol placement decides about final symbol recognition result. Symbol classification could not give good results if symbol location is not well done. Thus, both tasks are equally important in music symbols recognition.

Figure 2: Printed symbols of music notation - distortions, variety of fonts.

Since no universal music font exits, c.f. Figure 1, symbols of one class may have different forms. Also size of individual symbols does not keep fixed proportions. Even the same symbols may have different sizes in one score. Besides usual noise (printing defects, careless scanning) extra noise is generated by staff and ledger lines, densely packed symbols, conflicting placement of other symbols, etc.

A wide range of methods are applied in music symbol recognition: neural networks, statistical pattern recognition, clustering, classification trees, etc., c.f. (Bainbridge and Bell, 2001; Carter and Bacon, 1992; Fujinaga, 2001; Homenda and Mossakowski, 2004; McPherson, 2002). Classifiers are usually applied to a set of features representing processed symbols, c.f. (Homenda and Mossakowski, 2004). In next section we present application of neural networks as example classifier.

2.2 Neural Networks as Symbol Classifier

Having understood the computational principles of massively parallel interconnected simple neural processors, we may put them to good use in the design of practical systems. But neurocomputing architectures are successfully applicable to many real life problems. The single or multilayer fully connected feedforward or feedback networks can be used for character recognition, c.f. (Homenda and Luckner, 2004).

Experimental tests were targeted on classification of quarter, eight and sixteen rests, sharps, flats and naturals, c.f. Figure 2 for examples music symbols. To reduce dimensionality of the problem, the images were transformed to a space of 35 features. The method applied in feature construction was the simplest one, i.e. they were created by hand based on understanding of the problem being tackled. The list of features included the following parameters computed for bounding box of a symbol and for four quarters of bounding box spawned by symmetry axes of the bounding box:

- mean value of vertical projection,

- slope angle of a line approximating vertical projection,
- slope angle of a line approximating histogram of vertical projection;
- general horizontal moment m_{10},
- general vertical moment m_{01},
- general mixed moment m_{11}.

The following classifiers were utilized: backpropagation perceptron, feedforward counterpropagation maximum input network and feedforward counterpropagation closest weights network. An architecture of neural network is denoted by a triple input - hidden - output which identifies the numbers of neurons in input, hidden and output layers, respectively, and does not include bias inputs in input and hidden layers. The classification rate for three symbols on music notation: flats, sharps and naturals ranges between 89% and 99 9% , c.f. (Homenda and Mossakowski, 2004). Classifier applied: backpropagation perceptron, feedforward counterpropagation maximum input network and feedforward counterpropagation closest weights network. An architecture of neural network is denoted by a triple *input* - *hidden* - *output* which identifies the numbers of neurons in input, hidden and output layers, respectively, and does not include bias inputs in input and hidden layers.

3 REPRESENTING MUSIC INFORMATION

Acquired knowledge has to be represented and stored in a format *understandable* by the computer brain, i.e. by a computer program - this is a fundamental observation and it will be exploited as a subject of discussion in this section. Of course, a computer program cannot work without low level support - it uses a processor, memory, peripherals, etc., but they are nothing more than only primitive electronic tools and so they are not interesting from our point of view. Processing of such an acquired image of the paper document is a clue to the paper-to-memory data transfer and it is successfully solved for selected tasks, c.f. OCR technology. However, documents that are more complicated structurally than linear (printed) texts raise the problem of data aggregation in order to form structured space of information. Such documents raise the problem of acquiring of implicit information/knowledge that could be concluded from the relationships between information units. Documents containing graphics, maps, technical drawings, music notation, mathematical formulas, etc. can illustrate these aspects of difficulties of paper-to-computer-memory data flow. They are research subjects and still raise a challenge for software producers.

Optical music recognition (OMR) is considered as an example of paper-to-computer-memory data flow. This specific area of interest forces specific methods applied in data processing, but in principle, gives a perspective on the merit of the subject of knowledge processing. Data flow starts from a raster image of music notation and ends with an electronic format representing the information expressed by a scanned document, i.e. by music notation in our case. Several stages of data mining and data aggregation convert the chaotic ocean of raster data into shells of structured information that, in effect, transfer structured data into its abstraction - music knowledge. This process is firmly based on the nature of music notation and music knowledge. The global structure of music notation has to be acquired and the local information fitting this global structure must also be recovered from low level data. The recognition process identifies structural entities like staves, group them into higher level objects like systems, than it links staves of sequential systems creating instrumental parts. Music notation symbols very rarely exist as standalone objects. They almost exclusively belong to structural entities: staves, systems, parts, etc. So that the mined symbols are poured into these prepared containers - structural objects, cf. (Bainbridge and Bell, 2001; Dannenberg and Bell, 1993; Homenda, 2002; Taube, 1993). Music notation is a two dimensional language in which the importance of the geometrical and logical relationships between its symbols may be compared to the importance of the symbols alone. This phenomenon requires that the process of music knowledge acquisition must also be aimed at recovering the implicit information represented by the geometrical and logical relationships between the symbols and then at storing the recovered implicit relationships in an appropriate format of knowledge representation.

There are open problems of information gaining and representation like, for instance, performance style, timbre, tone-coloring, feeling, etc. These kinds of information are neither supported by music notation, not could be derived in a reasoning process. Such kinds of information are more subjectively perceived rather than objectively described. The problem of definition, representation and processing of ”subjective kinds of information” seems to be very interesting from research and practical perspectives. Similarly, problems like, for instance, human way of reading of music notation may be important from the

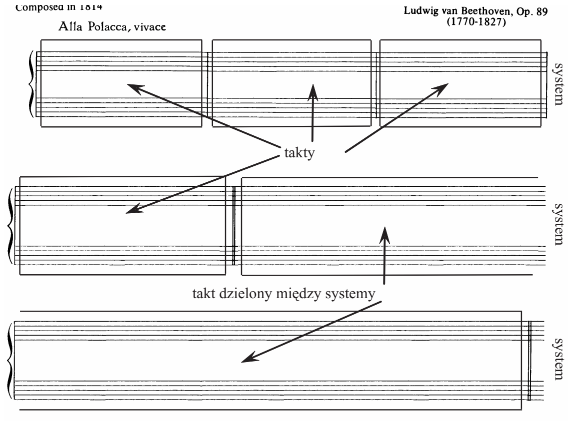

Figure 3: Structuring music notation - systems and measures.

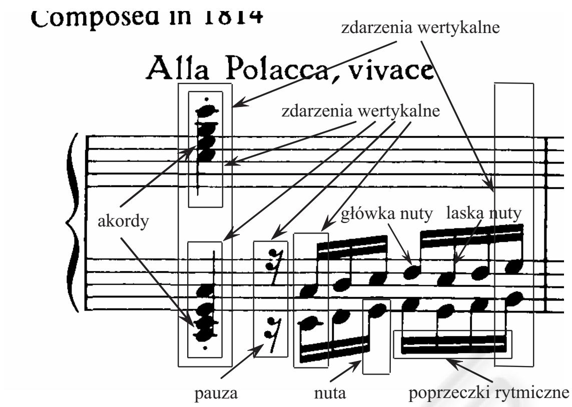

Figure 4: Structuring music notation - symbols, ensembles of symbols.

point of view of music score processing, cf. (Goolsby, 1994a; Goolsby, 1994b). Nevertheless, processing of such kinds of information does not fit framework of the paper and is not considered.

The process of paper-to-computer-memory music data flow is presented from the perspective of a paradigm of granular computing, cf. (Pedrycz, 2001). The low-level digitized data is an example of numeric data representation, operations on low-level data are numeric computing oriented. The transforming of a raster bitmap into compressed form, as e.g. run lengths of black and while pixels, is obviously a kind of numeric computing. It transfers data from its basic form to more compressed data. However, the next levels of the data aggregation hierarchy, e.g. finding the handles of horizontal lines, begins the process of data concentration that become embryonic knowledge units rather than more compressed data entities, cf. (Homenda, 2005).

3.1 Staves, Measures, Systems

Staves, systems, measures are basic concepts of music notation, cf. Figure 3. They define the structure of music notation and are considered as information quantities included into the data abstraction level of knowledge hierarchy. The following observations justify such a qualification.

A stave is an arrangement of parallel horizontal lines which together with the neighborhood are the locale for displaying musical symbols. It is a sort of vessel within a system into which musical symbols can be ”poured”. Music symbols, text and graphics that are displayed on it belong to one or more parts. Staves, though directly supported by low-level data, i.e. by a collection of black pixels, are complex geometrical shapes that represent units of abstract data. A knowledge unit describing a stave includes such geometrical information as the placement (vertical and horizontal) of its left and right ends, staff lines thickness, the distance between staff lines, skew factor, curvature, etc. Obviously, this is a complex quantity of data.

A system is a set of staves that are played in parallel; in printed music all of these staves are connected by a barline drawn through from one stave to next on their left end. Braces and/or brackets may be drawn in front of all or some of them.

A measure is usually a part of a system, sometimes a measure covers the whole system or is split between systems, cf. Figure 3. A measure is a unit of music identified by the time signature and rhythmic value of the music symbols of the measure. Thus, like in the above cases, a measure is also a concept of data abstraction level.

3.2 Notes, Chords, Vertical Events, Time Slices

Such symbols and concepts as notes, chords, vertical events, time slices are basic concepts of music notation, cf. Figure 4. They define the local meaning of music notation and are considered as information quantities included in the data abstraction level of the knowledge hierarchy. Below a description of selected music symbols and collections of symbols are described. Such a collection constitutes a unit of information that has common meaning for musician. These descriptions justify classification of symbols to in the data abstraction level of the knowledge hierarchy.

Note - a symbol of music notation - represents basically the tone of given time, pitch and duration. A note may consist of only a notehead (a whole note) or also have a stem and may also have flag(s) or beam(s). The components of a note are information quantities created at the data concentration and the data aggregation stages of data aggregation process. This components linked in the concept of a note create an abstract unit of information that is considered as a more complex component of the data abstraction level of the information hierarchy.

A chord is composed of several notes of the same duration with noteheads linked to the same stem (this description does not extend to whole notes due to the absence of a stem for such notes). Thus, a chord is considered as data abstraction.

A vertical event is the notion by which a specific point in time is identified in the system. Musical symbols representing simultaneous events of the same system are logically grouped within the same vertical event. Common vertical events are built of notes and/or rests.

A time slice is the notion by which a specific point in time is identified in the score. A time slice is a concept grouping vertical events of the score specified by a given point in time. Music notation symbols in a music representation file are physically grouped by page and staff, so symbols belonging to a common time slice may be physically separated in the file. In most cases this is time slice is split between separated parts for the scores of part type, i.e. for the scores with parts of each performer separated each of other. Since barline can be seen as a time reference point, time slices can be synchronized based on barline time reference points. This fact allows for localization of recognition timing errors to one measure and might be applied in error checking routine.

4 BRAILLE SCORE

Braille Score is a project developed and aimed on blind people. Building integrated music processing computer program directed to a broad range of blind people is the key aim of Braille Score, c.f. (Moniuszko, 2006). The program is mastered by a man. Both the man and computer program create an integrated system. The structure of the system is outlined in Figure 5.

The system would act in such fields as:

- creating scores from scratch,
- capturing existing music printings and converting them to electronic version,
- converting music to Braille and printing it automatically in this form,

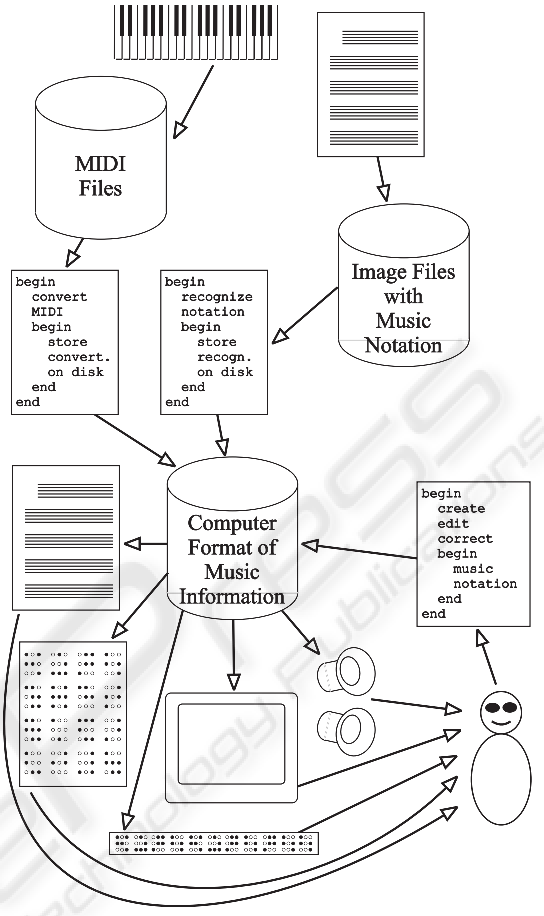

Figure 5: The structure of Braille Score.

- processing music: transposing music to different keys, extracting parts from given score, creating a score from given parts,
- creating and storing own compositions and instrumentations of musicians,
- a teacher's tool to prepare teaching materials,
- a pupil's tool to create their own music scores from scratch or adapt acquired music,
- a hobby tool.

4.1 User Interface Extensions for Blind People

Braille Score is addressed to blind people, c.f (Moniuszko, 2006). Its user interface extensions allow blind user to master the program and to perform operations on music information. Ability to read, edit and print music information in Braille format is the most important feature of Braille Score. Blind user is provided the following elements of interface: Braille notation editor, keyboard as input tool, sound communicator.

Blind people do not use pointing devices. In consequence, all input functions usually performed with mouse must be mapped to computer keyboard. Massive communication with usage of keyboard requires careful design of interface mapping to keyboard, c.f. (Moniuszko, 2006).

Blind user usually do not know printed music notation. Their perception of music notation is based on Braille music notation format presented at Braille display or punched sheet of paper, c.f. (Krolick, 1998). In such circumstances music information editing must be done on Braille music notation format. Since typical Braille display is only used as output device, such editing is usually done with keyboard as input device. In Braille Score Braille representation of music is converted online to internal representation and displayed in the form of music notation in usual form. This transparency will allow for controlling correctness and consistency of Braille representation, c.f. (Moniuszko, 2006).

Sound information is of height importance for blind user of computer program. Wide spectrum of visual information displayed on a screen for user with good eyesight could be replaced by sound information. Braille Score provides sound information of two types. The first type of sound information collaborates with screen readers, computer programs dedicated to blind people. Screen readers could read contents of a display and communicate it to user in the form of synthesized speech. This type of communication is supported by contemporary programming environments. For this purpose Braille Score uses tools provided by Microsoft .NET programming environment. The second type of sound information is based on own Braille Score tools. Braille Score has embedded mechanism of sound announcements based on its own library of recorded utterances.

5 CONCLUSIONS

The aim of this paper is a discussion on involvement of computational intelligence methods in implementation of user friendly computer programs focused on disabled people. In te paper we describe a concept of Braille Score the specialized computer program which should help blind people to deal with music and music notation. The use of computational intelligence tolls can improve the program part devoted to recognition and processing of music notation. The first results with Braille Score show its to be a practical and useful tool.

REFERENCES

Bainbridge, D. and Bell, T. (2001). The challenge of optical music recognition. *Computers and the Humanities*, 35:95–121.

Carter, N. P. and Bacon, R. A. (1992). *Automatic Recognition of Printed Music*, pages 456–465. in: Structured Document Analysis, Analysis, H.S.Baird, H.Bunke, K.Yamamoto (Eds). Springer Verlag.

Dannenberg, R. and Bell, T. (1993). Music representation issues, techniques, and systems. *Computer Music Journal*, 17(3):20–30.

Fujinaga, I. (2001). Adaptive optical music recognition. In *16th Inter. Congress of the Inter. Musicological Society, Oxford*.

Goolsby, T. W. (1994a). Eye movement in music reading: Effects of reading ability, notational complexity, and encounters. *Music Perception*, 12(1):77–96.

Goolsby, T. W. (1994b). Profiles of processing: Eye movements during sightreading. *Music Perception*, 12(1):97–123.

Homenda, W. (1996). Automatic recognition of printed music and its conversion into playable music data. *Control and Cybernetics*, 25(2):353–367.

Homenda, W. (2002). Granular computing as an abstraction of data aggregation - a view on optical music recognition. *Archives of Control Sciences*, 12(4):433–455.

Homenda, W. (2005). Optical music recognition: the case study of pattern recognition. In *in: Computer Recognition Systems, Kurzynski et al (Eds.)*, pages 835–842. Springer Verlag.

Homenda, W. and Luckner, M. (2004). Automatic recognition of music notation using neural networks. In *in: Proc. of the International Conference On Artificial Intelligence and Systems, Divnomorskoye, Russia, September 3-10*, pages 74–80, Moscow. Physmathlit.

Homenda, W. and Mossakowski, K. (2004). Music symbol recognition: Neural networks vs. statistical methods. In *in: EUROFUSE Workshop On Data And Knowledge Engineering, Warsaw, Poland, September 22 - 25*, pages 265–271, Warsaw. Physmathlit.

Krolick, B. (1998). *How to Read Braille Music*. Opus Technologies.

McPherson, J. R. (2002). Introducing feedback into an optical music recognition system. In *in: Third Internat. Conf. on Music Information Retrieval, Paris, France*.

Moniuszko, T. (2006). Design and implementation of music processing computer program for blind people (in polish). Master's thesis, Warsaw University of Technology.

Pedrycz, W. (2001). Granular computing: An introduction. In *in: Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver*, pages 1349–1354.

Taube, H. (1993). Stella: Persistent score representation and score editing in common music. *Computer Music Journal*, 17(3):38–50.