Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100105

Automatic Data Understanding: the Tool for Intelligent Man-Machine Communication

Wladyslaw Homenda[1,2,*] *and Witold Pedrycz*[3,4,5]

[1] Faculty of Mathematics and Information Science, Warsaw University of Technology, ul. Koszykowa 75, 00-662 Warsaw, Poland
[2] Faculty of Economics and Informatics in Vilnius, University of Bialystok, Kalvariju G. 135, LT-08221 Vilnius, Lithuania
[3] Department of Electrical & Computer Engineering, University of Alberta, Edmonton T6R 2G7 AB Canada
[4] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University Jeddah, 21589, Saudi Arabia
[5] Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, 01-447 Warsaw, Poland

Received: 8 Jun. 2015, Revised: 6 Aug. 2015, Accepted: 7 Aug. 2015
Published online: 1 Jan. 2016

Abstract: The paper is focused on man-machine communication, which is perceived in terms of data exchange. Understanding data being exchanged is the fundamental property of intelligent communication. The main objective of this paper is to introduce the paradigm of intelligent data understanding. The paradigm stems from syntactic and semantic characterization of data and is soundly based on the paradigm of granular structuring of data and computation. The paper does not introduce a formal theory of intelligent data understanding. Instead this paradigm as well as notions of granularity, semantics and syntax are cast within the domain of music information. The domain immersion is forced by a substantial dependence of details of the paradigm of automatic data understanding on application in a given domain.

Keywords: syntax, semantics, granularation, automatic data understanding, man-machine communication

1 Introduction

We assume that data, the subject of communication, create a structured space of information. Therefore communication requires identification of local and global structures of data and accomplishing operations on data and structures of data. The study includes discussion on syntactic description and semantic analysis of data and on granulation of syntax and semantics. The study on man-machine communication is reflected in the domain of paginated music notation. Some important structural operations give empirical view on automatic data understanding.

In this paper we do not develop a formal and universal theory of man-machine communication or automatic data understanding. We do not believe that such universal solution becomes possible at least now. Instead, besides formulation of notions of intelligent data understanding, granularity, semantics and syntax, we provide a case study in the domain of music information. The case study shows how to cast these notions onto the domain. Adaptation of the introduced ideas to other domains could be carried out in a similar fashion.

The paper is structured as follows. In consecutive sections of Section 2 we provide the meaning of basic terms used in the paper: syntax, semantics, granulation, intelligent understanding, case studies versus generalization, the subject of the study, languages of communication. Section 3 covers discussion on syntactic structuring of paginated music notation. Semantics and granulation reflected in paginated music notation is outlined in Section 4. The paradigm of data understanding is empirically presented in Section 5.

2 Concepts

Communication is a process of information exchange. Communication is accomplished in some language. For instance, composer communicates musician on how to perform in a given key uses the key signature, which is put in respective place of music notation, teacher instructs students in given natural language how to solve the problem. In all such cases information being exchanged is expressed and/or described as a text in some language and

* Corresponding author e-mail: homenda@mini.pw.edu.pl

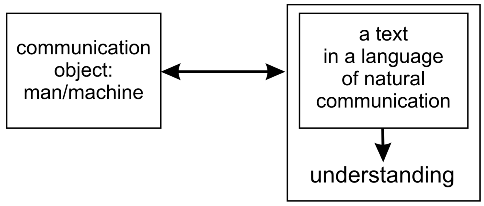

Fig. 1: An overall diagram of communication. Either man or machine is the object of communication.

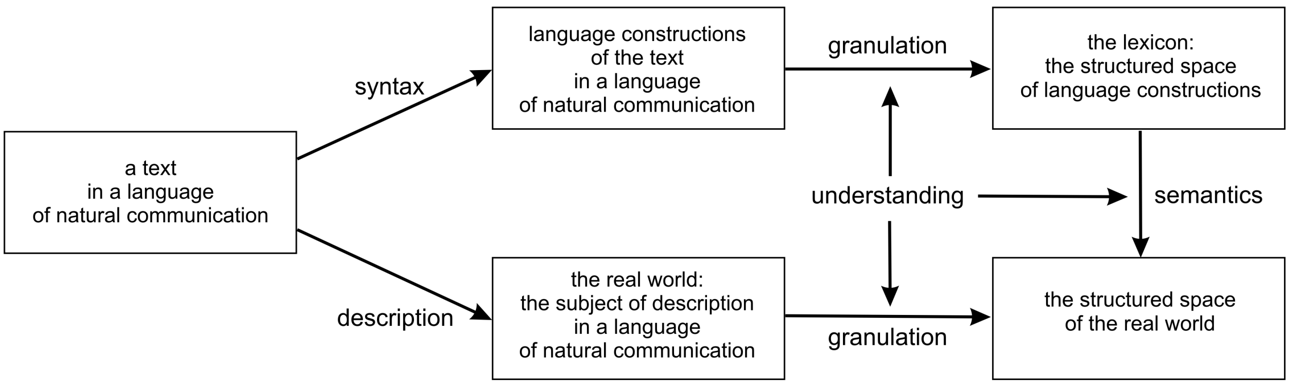

Fig. 2: An overall diagram of the paradigm of information understanding

intelligent communication is just exchanging texts expressing information and/or describing data. The general scheme of communication is shown in Figure 1. Communication seen as texts exchange must be supplemented by understanding data described by texts.

2.1 Syntax, semantics, granulation, understanding

In Figure 2 we illustrate the meaning of the paradigm of understanding. This paradigm is built on the basis of syntactic structuring, semantic analysis and granulation. Below we briefly describe meaning of these terms referring either to common terms, or to respective sources. It is worth stressing that in general we consider known concepts and ideas to integrate them into the novel approach.

We consider five key entities in the diagram of understanding. These entities are commonly understood, or are described in further sections of this paper:

– *the text in a language of natural communication*, this may be a text in a natural language, a score of paginated music notation, a score of Braille music notation, etc.,
– *language constructions* of the text in a language of natural communication, these are such items like texts and sentences of a natural language, parts of sentences, scores of music notation and their parts. It is assumed that these items are derivable in a grammar describing the language of natural communication,
– *the lexicon: the structured space of language constructions*, it is the space of language constructions, each of them supplemented with possible derivation (parsing) subtrees, the lexicon includes relations between items of this space,
– *the real world: the subject of description in a language of natural communication*, it is a world of things, sensations, thoughts, ideas etc., usually a part of this world is described by a text in a language of natural communication, sometimes the whole world is described by such a text,
– *the structured space of the real world*, it is the space of things sensations, thoughts, ideas etc. provided with structures based on relations between items and sets of items.

Syntax, i.e. *syntactic structuring* of texts of languages of natural communication (and also languages of formal communication) is the study on how words fit together structures describing a world of real objects, c.f. [7]. Syntactic structuring is usually described by a kind of a grammar. Such a description can be given in a form of a set of rules or as a formal grammar. Our study is based on context-free grammars, c.f. [7] for detailed justification of the choice of context-free methods.

Granulation is a concept of structuring of the space of language constructions and of the real world. Structuring the space of language constructions leads to

Fig. 3: Beginning excerpts of the Polonaise from Sonata for Flute and Piano by Beethoven (upper part) and the Suite (Ouverture) No. 3 in D major by J. S. Bach, BWV 1068, transcription for Piano by T. A. Johnson, the second movement Air (bottom part)

the formation of the lexicon. The concept of lexicon is elaborated on in [8]. Language structures supplemented by corresponding parts of derivation trees (parsing trees) create syntactic structures of the lexicon. Creation of the lexicon on the basis of language constructions is an apparent application of the paradigm of granularity, c.f. [13,14] for the granularity paradigm.

Semantics captures a relation between items of lexicon and items of the structured space of the real world. In other words, semantics casts the lexicon onto the structured space of the real world. In general, such relations/casts are many-to-many dependencies. Sometimes, this may be many-to-one or one-to-one relation, i.e. it may be a mapping form the lexicon to the structured space of the real world.

Understanding is an ability to construct granular structures in both spaces: the lexicon and the real world and to construct semantics.

2.2 Generalization versus case studies

Our approach to intelligent human-machine communication is focused on automatic understanding of data as its fundamental feature. The paradigm of automatic data understanding involves syntactic description, semantic analysis and granular structuring as general methods. However, we do not attempt to generalize these methods in a way that they would be directly applicable in any subject of communication. Instead, we provide a case study in a chosen domain. This approach would be adopted to other domains. In our opinion, the paradigm of automatic data understanding is the universal approach to intelligent human-machine communication. However, the details of such approach are domain dependent. Therefore, details discussed in this paper may not be universally applicable in any domain. An attempt to develop detailed methods of automatic data understanding, which would be domain independent, is not possible at the current level of technology.

2.3 The case study

In this paper, we analyze structured spaces of music data oriented to paginated music notation. The study is focused on operations done on spaces of music information. Methods presented in the paper are illustrated with fragments of scores of pieces of classical music. Namely, we inspect *Polonaise from Sonata for Flute and Piano* by L. v. Beethoven and *Suite No. 3 in D major* by J. S. Bach transcribed for Piano by T. A. Johnson, c.f. Figure 3. Our discussion embraces only elements of both scores. Nevertheless, the methods illustrated with these pieces can be expanded on the entire pieces as well as on other scores. Similarly, the study focused on spaces of music information oriented to paginated music notation may be adopted to other types of descriptions of music information. Examples of other descriptions of music information are Music XML cf. [2], Braille Music cf. [11], MIDI cf. [12].

2.4 Languages - tools of communication

For all the time communication between human beings has been done in some languages. Natural languages like English, Spanish, Russian, music notation, gesture language, technical drawings and C/C++/C# programming languages are examples of languages used in communication, c.f. [7]. Besides them we can consider tools of communication such as painting, sculpture, music etc. It is easily seen that tools of communication mentioned above can be split to three groups:

– languages of natural communication
– languages of formal communication
– languages of arts

Below we briefly describe these groups of languages without detailed characterization of them.

The group of ***languages of natural communication*** includes natural languages, music notation, gesture language etc. The common feature of languages of natural communication is that they had been created, developed and used prior to their formal codification and - up to now - they have no full formal definition. Moreover, they are used intuitively with high-level of flexibility. What is more, some kind of anarchy, mistakes and errors in usage of languages of natural communication does not break communication and is not even harmful for successful data exchange.

The group of ***languages of formal communication*** is represented by technical drawings, programming languages and systems of menus, toolbars and dialog boxes. Unlike in the case of the first group, languages of formal communication were developed and fully defined prior to their usage. For languages of formal communication no anarchy in their usage is admitted. Also, they are hardly tolerant of mistakes and errors.

The group of ***languages of arts*** includes painting, sculpture, performed music, dance etc. These languages are not considered in our study.

3 Syntactic approach

Syntactic approach forms a crucial stage and a crucial problem in the wide spectrum of tasks as, for instance, pattern recognition, translation of programming languages, processing of natural languages, music processing, etc. By syntactic approach and syntactic methods we understand grammars, automata, algorithms used in processing languages.

By analogy to the Chomsky's hierarchy of languages, syntactic approaches and methods can be categorized as regular, context-free and context-sensitive, c.f. [9].

In this paper, we use syntactic methods to describe languages of natural communication. Unfortunately, Chomsky's taxonomy of languages of natural communication is not known, what means that every language of natural communication should be individually classified. It is even not known, if natural languages are context-free or context-sensitive. According to [10], pp. 488-491, many attempts to demonstrate that English is a context-sensitive language and not context-free one have occurred either to be unsuccessful or incorrect.

As to the paginated music notation, this notation seems to be context-sensitive language. It contains repetitions (i.e. constructions of the forms $ww, www, wwww, \ldots$ such that w is a part of paginated music notation), which are context sensitive. Repetitions in a score must be synchronized in time, so then this is an example of such constructions. Part type scores, i.e. scores split to parts designated to musicians playing given instruments, is another type of a such constructions.

3.1 Context-free syntactic description

The Chomsky's taxonomy of languages of natural communication is of less importance for practical applications. Even if such languages are context-free ones, they are too complex to be fully formalized, c.f. [1]. Despite that there is a definite set of rules defining a language of natural communication, the rules are much more complicated than, for instance, rules describing languages of formal communication. And such rules can often be broken with little impact on communication. Thus, a description of a language of natural communication must definitely be highly flexible and deeply tolerant to natural anarchy of its subjects. With this in mind, the proposed approach to describing languages of natural communication will rely on the sensible application of context-free methods applied locally or on covering languages (i.e. generating all constructions of a language and constructions not belonging to it or incorrect constructions of the language). Moreover, we assume that the context-free methods will not be applied unfairly to generate incorrect constructions or constructions not belonging to them. These assumptions allow only for a raw approximation of languages of natural communication. Of course, such assumptions are real shortcomings when comparing to an accurate description. The shortcomings must be solved by employing some other methods.

3.1.1 The tool

The discussion on describing paginated music notation is based on common definition of grammars and of context-free grammars. We assume that the reader is familiar with the basic notions of mathematical linguistic. Therefore, we only recall these basic notions.

Let us recall that a system $G = (V,T,P,S)$ is a grammar, where: (a) V is a finite set of *variables* (called also *nonterminals*), (b) T is a finite set of terminal symbols (simply called *terminals*), (c) a nonterminal S is the initial symbol of the grammar and (d) P is a finite set of productions. A pair (α,β) of strings of nonterminals and terminals is a production assuming that the first element α of the pair is a nonempty string. Productions are usually denoted $\alpha \rightarrow \beta$. Grammars having all productions with α being a nonterminal symbols are context-free grammars.

A derivation in a grammar is a finite sequence of strings of nonterminals and terminals such that: (a) the first string in this sequence is just the initial symbol of the grammar and (b) for any two consecutive strings in the sequence, the later one is obtained from the former one applying a production in the usual way, i.e. by replacing a substring of the former string equal to the left hand side of the production with the right hand side of the production. We say that the last element of the string is *derivable* in the grammar.

For a context-free grammar a derivation can be outlined in a form of derivation tree, i.e. (a) the root of the tree is labelled with the initial symbol of the grammar and (b) for any internal vertex labelled by the left side of a production, its children are labelled by symbols of the right side of the production.

3.1.2 The application

Under the assumptions of Section 3.1, usage of a simplified context-free grammar for the purpose of syntactical structuring of paginated music notation is valid in practice. The grammar will be applied for analysis of constructions, which are assumed to be well grounded pieces of paginated music notation. Of course, such a grammar can neither be applied in checking correctness of constructions of paginated music notation, nor in generation of such constructions.

In Table 1 we present a raw description of paginated music notation in a form of productions of a context-free grammar. The productions were constructed manually at the basis of observation of structures of paginated music notation. Paginated music notation is a collection of staves (staff lines) placed on pages. Every stave is surrounded by corresponding symbols placed on the stave and around it. The collection of staves has its own structure with staves grouped into higher-level units called systems. A raw description of the structure of music notation could be approximated by the set of context-free productions given below. The components of the grammar $G = (V,T,P,S)$ are as follows. The set of nonterminals includes all identifiers shown in triangle brackets printed in italic. The nonterminal $\langle score \rangle$ is the initial symbol of G. The set of terminal includes all non bracketed identifiers.

Music notation can be described by different grammars. Construction of such grammars may reflect various aspects of music notation, e.g. geometrical or logical structuring, c.f. [5,7]. The above description is constructed from the point of view of geometrical properties of music notation. The nonterminals $\langle page \rangle$ and $\langle system \rangle$ define items of music notation strictly related to paginated music notation. The nonterminal $\langle vertical\ event \rangle$ defines items related to paginated music notation, but it also plays important role as a logical element of music information.

The grammar presented here is a simplified version of a real grammar fully describing paginated music notation. Due to limitation of the paper, many details are skipped in this grammar. For instance, pitch and duration of notes, articulation and ornamentation symbols, dynamics, placement of graphical elements on the page etc. are not outlined. On the other hand, expansion of the grammar is not difficult in its essence and could be easily done based on this study.

Table 1: A grammar generating a piece of printed music notation

<score>	→	<score_part> <score> | <score_part>
<score_part>	→	<page> <score_part> | <page>
<page>	→	<system> <page> | <system>
<system>	→	<stave> <system> | <stave>
<system>	→	<part_name> <stave> <system> | <part_name> <stave>
<part_name>	→	Flute | Piano | *etc.*
<stave>	→	beginning-barline <bl_stave> | <clef> <bl_stave>
<stave>	→	beginning-barline <clef> <bl_stave> | <bl_stave>
<bl_stave>	→	<key_signature> <ks_stave> | <ks_stave>
<ks_stave>	→	<time_signature> <ts_stave> | <ts_stave>
<ts_stave>	→	<measure> <barline> <ts_stave> | <measure> <barline>
<measure>	→	<change_of_k_sign.> <ks_measure> | <ks_measure>
<ks_measure>	→	<change_of_t_sign.> <ts_measure> | <ts_measure>
<ts_measure>	→	<vertical event> <ts_measure> | <vertical event>
<vertical event>	→	<stem> <vertical event> | <stem>
<stem>	→	<beams> <note_stem> | <flags> <note_stem> | <note_stem>
<beams>	→	left-beam | right-beam | left-beam right-beam
<clef>	→	treble_clef | bass_clef
<flags>	→	flag <flags> | flag
<note_stem>	→	note-head <note_stem> | note-head

3.2 The lexicon

Lexicon is the key concept of granulation of syntax. Lexicon is a set of language constructions, which describe objects and local and global structures of objects in the real world. Elements of lexicon are phrases generated in a grammar supplemented with parts of derivation tree build on them. Such a part of the derivation tree, that corresponds to the given phrase, should be the minimal one covering the phrase. This part has its upper vertex, which is its root. There is the path connecting the root of the subtree to the root of the whole derivation tree. Extensions of the subtree along this path create more lexicon constructions based on the same phrase. Examples given in Figure 4 illustrate the concept of lexicon. All four elements of the lexicon are built on the same phrase of music notation, i.e. on the eight note. Part a) of the Figure shows the lexicon element with the minimal part of derivation tree. This lexicon element supplemented with two vertices and edges creates another lexicon element shown in part b) of this Figure. Parts c) and d) of this Figure show two more elements of the lexicon. Note that all these elements are built on the same phrase of music notation.

It is worth to recall that according to discussion in Section 3.1 the above grammar generates all valid paginated music notation as well as constructions not being correct music notations. However, we assume that only valid phrases of music notation will be subjected to analysis.

4 Semantics and granulation

It is clear that the context-free grammar of music notation defined in Section 3.1.2 is a tool used to describe music notation rather than a subject of communication and understanding. Therefore, the syntactic approach to describing the music notation, as expressed in the form of grammar given in Section 3.1.2, is a workout of a space of information onto constructions generated by context-free grammars. Such constructions describe music notation: notes, rests, vertical events, voice lines, measures, staves, systems, scores. On the other hand, music notation is context-sensitive, as noted in Section 3. For that reason, the study on human communication in this area, if restricted to the pure syntactic approach based on context-free methods, will not give a complete perspective on the information being exchanged. So then, we consider switching to mutual utilization of syntactic and semantic methods.

4.1 Valuation relation

Lexicon elements describe objects in a real world. In the case of music information the real world includes sounds produced when a piece of music is being performed. For simplicity, we assume that sounds are of the form of notes described by three parameters: pitch, beginning time and duration of a note. A triple describing a sound/note roughly can be supplemented by other features, e.g. volume (loudness), articulation (legato, staccato, portato etc.)

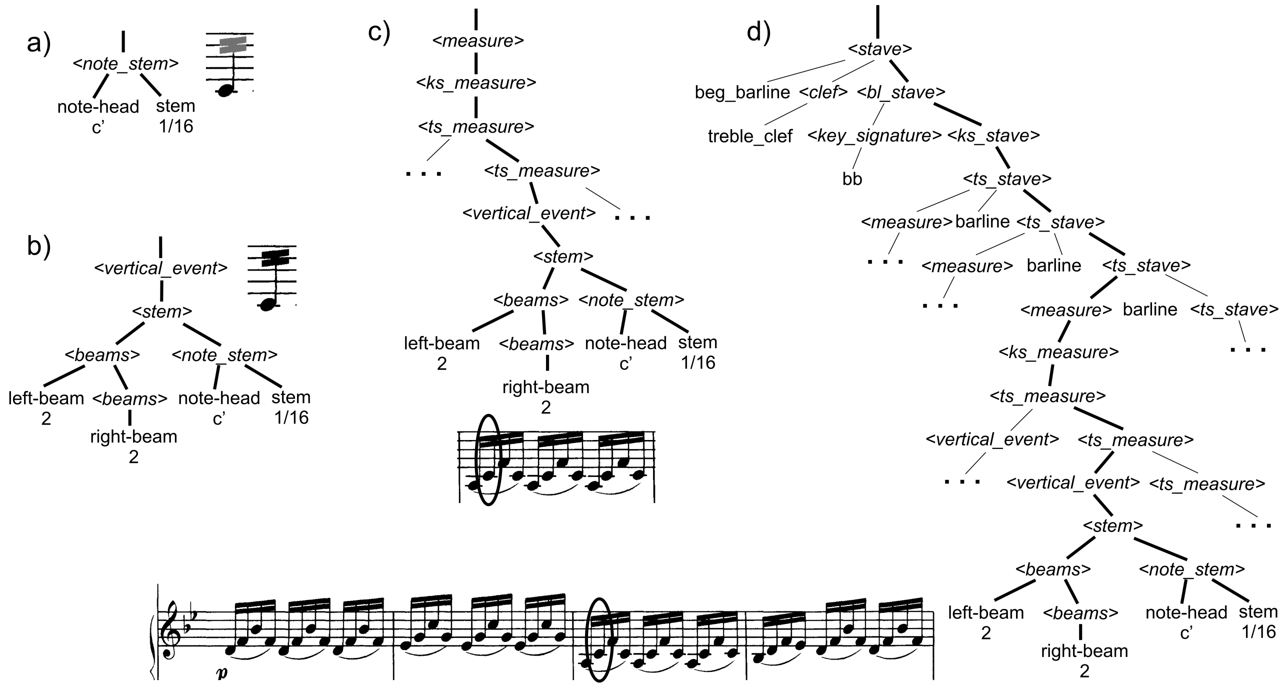

Fig. 4: Lexicon elements and their interpretation: a) a sixteenth note c', b) a sixteenth note c' twice beamed both ways, c) the sixteenth note c' twice beamed both ways, the second note in a measure d) the sixteenth note c' twice beamed both ways, the second note in the the third measure of the stave. We say that all these lexicon elements are hanged to *<note_stem>* node. We say that these elements are hooked in *<note_stem>*, *<vertical_event>*, *<measure>*, *<stave>* nodes, respectively. The part of the derivation tree of the Beethoven's score associated with the phrase (the sixteenth c' note) is indicated by thick edges. Extra nodes and edges are added for the sake of clarity.

Semantics, as interpreted in this paper, immerses the lexicon in the space of sounds/notes, cf. Figure 5. Such the immersion is a relation defined in the Cartesian product of lexicon and the space of sounds/notes. Note that lexicon elements shown in Figure 4 may define many objects in the world. If the root of the lexicon element is hooked in the root of the derivation tree, ten such lexicon element defines unique object in the score. From geometrical perspective, semantics of the case a) defines all sixteenth note placed at the second space of the stave. Considering logical perspective, since selected note is in scope of treble clef, semantics defines notes c' in the score, i.e. it defines one-lined c sixteenth notes having different beginning times. In cases b) we have notes double beamed both ways. In case c) we have notes placed in the given context, i.e. context defined by position of the note in a measure. In case d) only third measure of a stave is considered. If the lexicon element begins with the root of the derivation tree, then semantics defines the unique element: the note with given beginning time and given pitch.

Summarizing - semantics is a relation V in the lexicon L and the world W, called *valuation relation*:

$$V \subset L \times W$$

The world W includes objects and sets of objects described in a given language. In fact, objects of the world W usually create meaningful sets and - moreover - create structures with internal relations between objects. Such structures are also described by constructions of the language. Therefore, the world W is a structured space of objects (or structured space of information). Namely, the world W includes notes/sounds related to each other according to a given piece of music.

4.2 Granular space formation

The processing of music notation, as described in Section 3, leads to extracting information from a score. A collection of data items is subjected to a process of mining conceptual entities of information. This process finally leads to an overall understanding of the processed score. The process of structuring the space of information fits the paradigms of granular computing and information granulation. Granular computing paradigm *raised from a need to comprehend the problem and provide a better insight into its essence rather than get buried in all unnecessary details. In this sense, granulation serves as an abstraction mechanism that reduces an entire conceptual burden. As a matter of fact, by changing the*

Fig. 5: An excerpt of printed music notation (upper part) and its reflection in the space of sounds (bottom part). The space of sounds is illustrated in MIDI-like method. The valuation relation is shown in a few notation constructions and corresponding sounds.

size of the information granules, we can hide or reveal a certain amount of details one intends to deal with during a certain design phase, cf. [13]. It is worth stressing that information granules *not only support conversion of clouds of detailed data into more tangible information granules but, very importantly, afford a vehicle of abstraction that allows to think of granules as different conceptual entities*, see [13,14] and the references therein.

The description of music notation as well as music notation itself could be innately subjected to the paradigm of granular computing elucidation. As stated in [13], granular computing as opposed to numeric computing is knowledge-oriented. Information granules exhibit different levels of knowledge abstraction, what strictly corresponds to different levels of granularity. Depending upon the problem at hand, we usually group granules of similar size (i.e. similar granularity) together into a single layer. If more detailed (and computationally intensive) processing is required, smaller information granules are sought. Then, those granules are arranged in another layer. In total, the arrangement of this nature gives rise to the information pyramid. In the granular processing we encounter a number of conceptual and algorithmic layers indexed by the *size* of information granules. Information granularity implies the usage of various techniques that are relevant for the specific level of granularity.

4.2.1 Sizing syntactic granules

The meaning of granule size is defined accordingly to real application and should be consistent with commonsense and with the knowledge base. The intuition says that bigger the description, bigger the size of it. This intuition, compatible with commonsense, should be satisfied in sizing syntactic granules.

Let us recall that syntactic granules are simply elements of the lexicon. Therefore, we should describe size of lexicon elements. It can be done in terms of the number of nodes in the associated subtree. There are two aspects of lexicon elements affecting commonsense meaning of this concept of size. The first aspect is related to depth of the root of the subtree covering the phrase: deeper the root in the derivation tree, smaller the size. The second aspect is related to the length of path attached to the root of the subtree: shorter the length, smaller the size.

In the first case, for fixed derivation tree, i.e. for given piece of music and given grammar, height of the subtree is invertible to depth of its root. Let us recall that depth of a node in the tree is defined as length of the path from the root of the tree to this node. Of course, number of nodes depends increasingly on the height of the subtree: higher the subtree, higher the number of nodes in it. Comparing indicated nodes in Figure 6 we can see that height and size of subtrees hanged on <*stem*> nodes are the same while height and size of the subtree hanged on the <*ts_measure*> are bigger than former ones.

In the second case, dependency is quite obvious: longer path attached to the root of the subtree, more nodes we get. In Figure 4 we can see that height of subtrees (with attached path to the root) increase going from the case a) to the case d).

4.2.2 Sizing semantic granules

Size of semantic granule is estimated as a quantity of real world objects. Referring to granules outlined in Figure 4, and to discussion in Section 4.1, we see that sizes of semantic granules defined by lexicon elements a) - d) are reversible to sizes of corresponding syntactic granules. On the other hand, recalling Figure 6, we easily see that sizes of semantic granules corresponding to syntactic granules based on nodes <*stem*> are smaller than the semantic granule defined by the syntactic one based on on the node <*ts_measure*>. For the sake of clarity, the later comparison assumes syntactic granules hooked in the root

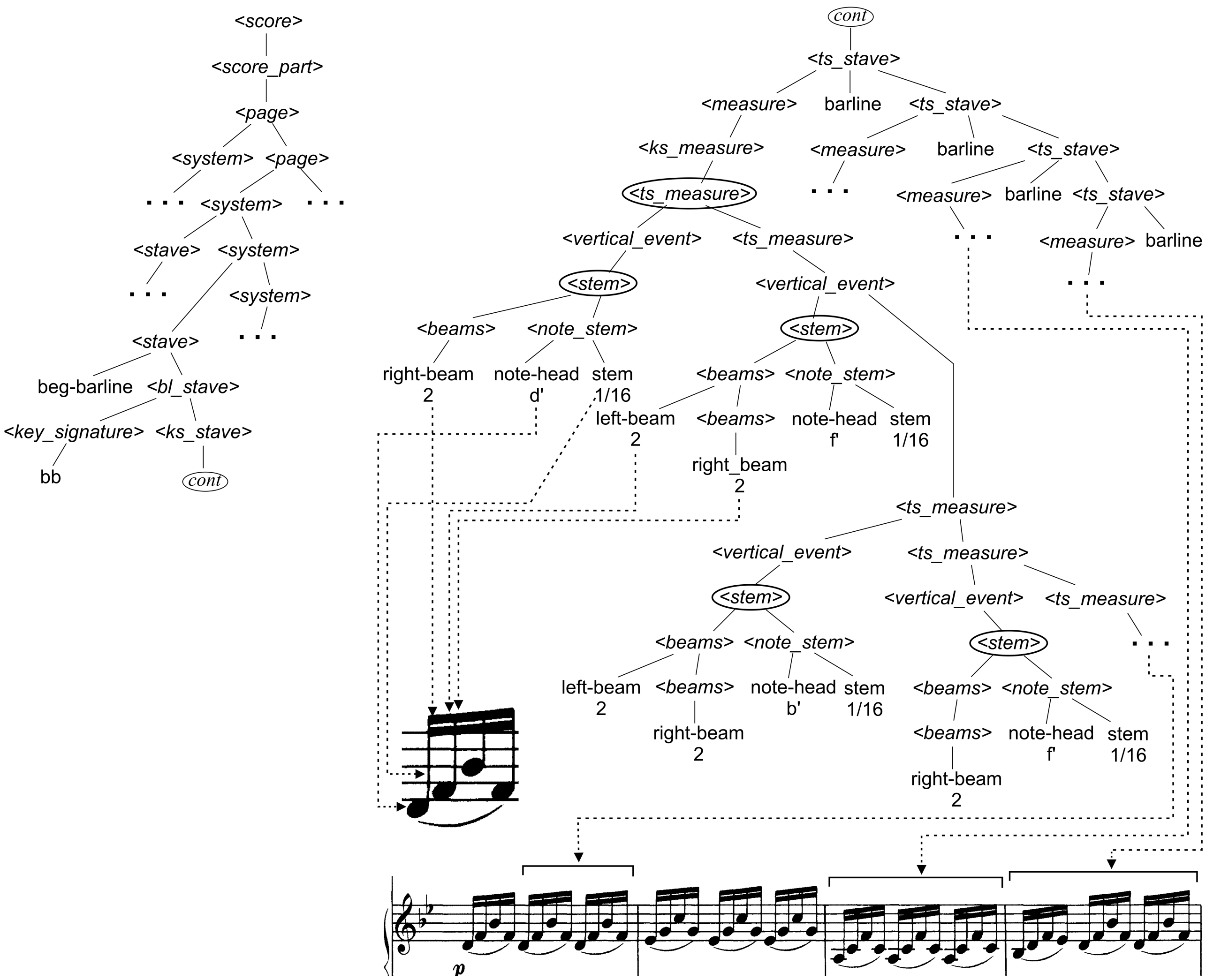

Fig. 6: A fragment of the derivation tree of *Polonaise from Sonata for Flute and Piano* by L. v. Beethoven. Derivation of the first beamed group in the second stave of the second system is expanded. Ellipsis indicate other parts of the score. The tree includes attributes of terminals: number of beams for left-beam and right-beam, pitch for head and duration for stem. Nodes *<ts_measure>* and *<stem>* are indicated for the discussion in Section 4.2.

of the whole derivation tree. The detailed discussion concerning syntactic granules other than hooked to the root is not presented in this study.

Amazingly, greater size of syntactic granule correspond to smaller size of respective semantic granule. The relevance between syntactic and semantic granules is shown in Figure 7. And, as in music notation case, this relevance is a manifestation of duality phenomenon in syntax-semantics related spaces: a) the *small* lexicon element defines sixteenth notes c’ in the whole score, b) bigger lexicon element defines both way double beamed sixteenth notes c’ in the whole score, c) next bigger lexicon element defines both way double beamed sixteenth notes c’ in given place of any measure, d) next bigger lexicon element defines both way double beamed sixteenth notes c’ in given place of the third measure of any stave, e) relevance of syntactic and semantic granules’ pyramids.

5 Understanding

As described in Section 2.1, data understanding is an ability to recognize semantics and granulation of information. According to discussion in Section 2.2 we do not attempt to develop a theoretical framework for automatic data understanding. Instead, we believe that case studies will be more informative in context of this paper. In subsequent sections we discuss examples of structural operations in the space of music information.

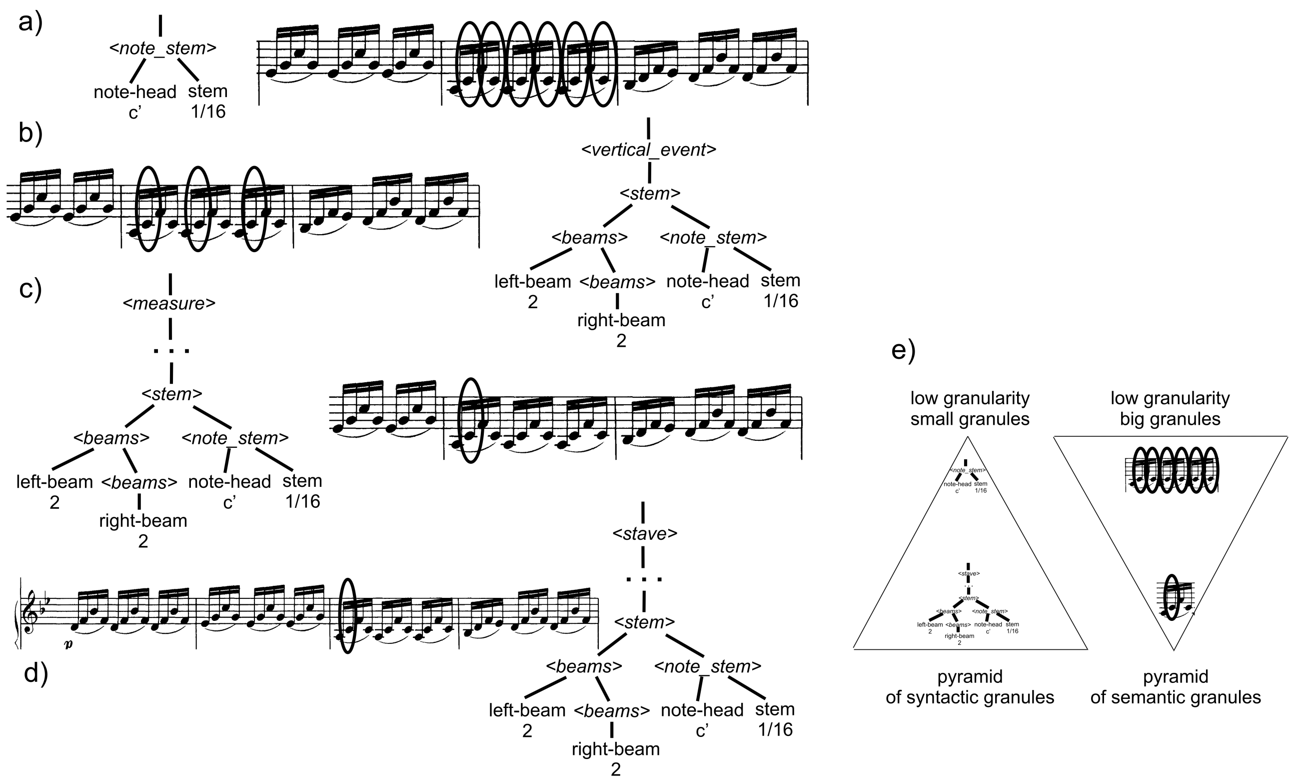

Fig. 7: Duality of syntax and semantics for music notation.

Such operations are the best manifestation of what we mean as data understanding.

5.1 Selecting

Among structural operations selection is the very fundamental one. Selection is the basis for other common structural operations like copying, finding, replacing etc. Selection is also the basis for domain specific operations. In case of paginated music notation, we can list such domain specific operations like transposition, conversion, harmonization, voice line identification etc.

Technically, drawing a rectangle with mouse or marking a sequence of symbols with keyboard or mouse is the most popular methods of selection. Possibly a multi selection can be done, usually with mouse moves or clicks while the key Ctrl is pressed.

This common operation in computer applications is usually performed on raw, low-level data. Examples of low level data selections are texts in text editors and regions of raster images (rectangles of pixels), c.f. [5,6]. An example of a raster image selection is shown in Figure 8: a rectangular region of the screen is selected and highlighted. This selection is interpreted as a part of a raster image and not the second and third measures of the upper stave of the paginated music notation. In this interpretation content of the image is of minor importance: it may be a part of printed music notation as well as a part of city map or a part of any other raster image.

In context of our study, such a selection should not be treated as a part of a screen and represented as a region of a raster bitmap. It should be considered as a part of printed music notation rather than a displayed image. Such selections can be characterized by lexicon elements and then - via valuation relation V - by a corresponding structure in the real world.

A selection is usually related to many elements of the lexicon. In Figure 4 the indicated sixteenth note can be selected simply by dragging a bounding box. However, such the selection may not have the unique interpretation, as it is outlined in this Figure. As a consequence, an interpretation of the selection affects its semantics. This uncertainty is not unexpected. Communication between people may also raise such situations, which are resolved with additional information. Alike, an extra tool supporting selection is required to fix such ambiguities.

Mouse-dragged rectangle is a simple form of selection. In complex spaces of information such the method of selection is far inadequate to needs. Music spaces of information are perfect examples of needs of more complex selection methods. An example of such selections is clarified in Figure 9. There are three voice

Fig. 8: An example of selection: a rectangle is drawn with the mouse.

Fig. 9: J. S. Bach, Suite (Ouverture) No. 3 in D major, BWV 1068, identification of voice lines. Three voice lines of the upper stave of the upper system are split to three separate staves in bottom system.

lines in the first stave of the Bach's score. In lower part of this Figure three voice lines are displayed in separate staves. It is clear that a rectangle dragging tool cannot select a voice line. This is why a more suitable tool should be used for such selections.

From human's perspective, a description in natural language would be the best way for describing such selection. For instance, the instruction *Select upper voice line in the upper stave of three beginning measures* clearly describes the selection. Analysis of such instructions opens a discussion on processing natural languages, which is not intended in this paper. Instead, for the sake of this study, we assume that instructions are formulated with simpler tools like mouse, keyboard and dialog boxes. For instance, we can use rectangle selection tool in the voice line mode to select voice lines.

Fig. 10: Illustration of searching. The pattern is in dotted ellipse, instances are dashed around. The upper part illustrates searching for exact matches (the same pitches, durations and time intervals between notes) without specific placement in the measure. The bottom part shows instances preserving pitch intervals, time intervals, durations and placement in the measure.

5.2 Searching

Searching is an operation of locating object(s) matching a given pattern, i.e. locating instance(s) identical or similar to the pattern. In text editors obvious meaning of searching operation is finding instances of a given string and no analysis of information is done. Searching tools admit more sophisticated methods of identification of instances. For example, regular expressions allow for finding strings of a language defined by such expression. Anyway, this is still the operation performed on strings and not on information brought by such strings. On the other hand, searching in raster bitmaps has no reasonable meaning unless content of the image is involved.

In context of this paper searching is performed on data structures rather than on such raw data as strings of symbols or region of rater bitmaps. The operation *Search* in a space of music information concerns a pattern, which is a part of space of music information. Searched pattern can be defined in different ways. Selection done as described in Section 5.1 is the simplest method for defining the search pattern.

Let us discuss the operation *Search* with the following example based on Figure 4. We assume that the *sixteenth c'* note is the pattern selected. We also assume that searching is accomplished along the piece with regard to pitch and duration. If the selection is interpreted as in the case a) of this Figure, i.e. *a sixteenth c'*, then any *sixteenth c'* note matches this pattern. This case is symbolically shown in Figure 7, case a), where any note c' matches the pattern. The pattern interpreted as in cases b) of Figure 4 defines the sixteenth note c' beamed twice both ways. In this case a sixteenth note c' matches the pattern assuming that is beamed both ways. Case c) defines any sixteenth note c' beamed both ways and with given placement in a measure. Finally, in case d), only notes described in case c) belonging to the third measure of a stave are admitted.

Let us consider another example based on Figure 10. Let us assume that the first beamed group of four sixteenths of the first measure is selected as the searching pattern and that only minimal part of derivation tree creates the corresponding lexicon element, i.e. the lexicon element is hooked in its hang. This pattern matches next two beamed groups of sixteenths in the same measure and last two beamed groups of sixteenths in the fourth measure of the same stave, c.f. Figure 10. If the lexicon element is hooked in the node <*measure*>, c.f. 6, then the pattern *the first beamed group of four sixteenths in a measure* does not match any instance. However, if we admit notes to be moved up or down by the same interval, then the pattern matches the first beamed group of four sixteenths in the third measure: notes of the instance are moved down by three halftones (semitone) comparing to the pattern, c.f. lower part of Figure 10. Of course, durations and time successions are preserved in all instances of both cases.

6 Conclusions

In this paper we introduced the paradigm of automatic data understanding. The paradigm stems from syntactic and semantic characterization of data and is soundly based on the paradigm of granular structuring of data and granular computing. We show the ways of employing syntactic structuring and semantic analysis in knowledge structuring and understanding data. The discussion is focused in the domain of music information as well as concepts are cast on the structured space of music information. The domain immersion is forced by a heavy dependence of details of the paradigm of automatic data understanding on application in a given domain. Although we do not introduce a formal theory of automatic data understanding, the study points out the methodology of knowledge processing, which can be applicable to different domains.

The main objective of this paper, i.e., automatic data understanding, is illustrated by several representative examples. The discussion outlines the paradigm of automatic data understanding as inherent feature of intelligent human-machine communication.

Acknowledgement

The research is supported by the National Science Center, grant No 2011/01/B/ST6/06478, decision no DEC-2011/01/B/ST6/06478.

References

[1] Beardon C., Lumsden D., Holmes G., Natural Language and Computational Linguistics, an introduction, New York, 1991.

[2] Castan G., Good M. and Roland P., Extensible Markup Language (XML) for Music Applications: An Introduction, W. B. Hewlett and E. Selfridge-Field (Eds.) The Virtual Score: Representation, Retrieval, Restoration, The MIT Press, 2001.

[3] Custodero, L. A., Construction of musical understandings: The cognition-flow interface, Bulletin for the Council of Research in Music Education, **142**, 7980 (1999).

[4] Hayes G. R. & Truong K. N., Paratyping: A Contextualized Method of Inquiry for Understanding Perceptions of Mobile and Ubiquitous Computing Technologies, Human-Computer Interaction, **28:3**, 265-286 (2013).

[5] Homenda W., Rybnik M., Querying in Spaces of Music Information, Proceedings of the IUKM, Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making, LNAI **7027**, pp. 243-255 (2011).

[6] Homenda W., Automatic data understanding: a necessity of intelligent communication, Rutkowski L. et al. (Eds.), LNAI **6114**, 476-483 (2010).

[7] Homenda W., Integrated syntactic and semantic data structuring as an abstraction of intelligent man-machine communication, Proc. of the ICAART - International Conf. on Agents and Artificial Intelligence, 324-330 (2009).

[8] Homenda W., Towards automation of data understanding: integration of syntax and semantics into granular structures of data, Proc. of the Fourth Int. Conf. on Modeling Decisions for Artificial Intelligence, MDAI **2007**, pp. 134-145 (2007).

[9] Hopcroft J. E., Ullman J. D., Introduction to Automata Theory, Languages and Computation, Addison-Wesley Publishing Company, Reading, Massachusetts, 1979, 2001.

[10] Jurafsky D., Martin J. H., Speech and Language Processing, Prentice Hall Inc., 2000.

[11] Krolick B., How to Read Braille Music, 2nd Edition, Opus Technologies, 1998.

[12] MIDI 1.0, Detailed Specification, Document ver. 4.1.1, February 1990.

[13] Pedrycz W., Granular Computing: An introduction, Proc. of the Joint 9th IFSA World Congress and 20th NAFIPS, Vancouver, Canada, 2001.

[14] Pedrycz W., Bargiela, A., Granular clustering: A granular signature of data, IEEE Trans. Syst. Man And Cybernetics - part B, **32(2)**, 212-224 (2002).

[15] Su N. M., Brdiczka O. & Begole B., The Routineness of Routines: Measuring Rhythms of Media Interaction, Human-Computer Interaction, **28:4**, 287-334 (2013).

[16] Tadeusiewicz R., Ogiela M. R., Automatic Image Understanding - A New Paradigm for Intelligent Medical Image Analysis, Bioalgorithms and Med-Systems, **2(3)**, 5-11 2006.

[17] Tadeusiewicz R., Ogiela M. R., Why Automatic Understanding?, LNCS, Springer-Verlag Berlin Heidelberg, **4432**, 477-491 (2007).

[18] Yardi, S., & Poole, E. S., Please help! Patterns of personalization in an online tech support board. C&T 09: Proc. of the Fourth International Conf. on Communities and Technologies. New York, NY: ACM, 2009.

Wladyslaw Homenda is an Associate Professor in the Faculty of Mathematics and Information Science at the Warsaw University of Technology, Warsaw, Poland. He is also with the Faculty of Economics and Informatics in Vilnius, Lithuania, at the University of Bialystok, Poland. His main research interests are in theoretical foundations of computer science, knowledge representation and processing and intelligent computing technologies, specifically in the areas of man-machine communication and human-centric computing, fuzzy modeling and Granular Computing, knowledge discovery and data mining. He currently serves as an Associate Editor of Information Sciences and is a member of several editorial boards of other international journals.

Witold Pedrycz is Professor and Canada Research Chair (CRC - Computational Intelligence) in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. His main research directions involve Computational Intelligence, fuzzy modeling and Granular Computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and Software Engineering. He has published numerous papers in this area. Prof. W. Pedrycz is intensively involved in editorial activities. He is an Editor-in-Chief of Information Sciences and served as Editor in Chief IEEE Trans. on Systems, Man, and Cybernetics - Systems. He currently serves as an Associate Editor of IEEE Trans. on Fuzzy Systems and is a member of a number of editorial boards of other international journals.