The Qualitative Report

Volume 26 | Number 2

How To Article 1

2-1-2021

Analytical Coding: Performing Qualitative Data Analysis Based on Programming Principles

Gennady Kanygin
Sociological Institute of the Russian Academy of Sciences, Branch of the Federal Center of Theoretical and Applied Sociology of the Russian Academy of Sciences, g.kanygin@gmail.com

Viktoria Koretckaia
Sociological Institute of the Russian Academy of Sciences, Branch of the Federal Center of Theoretical and Applied Sociology of the Russian Academy of Sciences, interperfection@gmail.com

Recommended APA Citation

Kanygin, G., & Koretckaia, V. (2021). Analytical Coding: Performing Qualitative Data Analysis Based on Programming Principles. *The Qualitative Report*, *26*(2), 316-333. https://doi.org/10.46743/2160-3715/2021.4342

Qualitative Research Graduate Certificate
Indulge in Culture
Exclusively Online • 18 Credits
LEARN MORE
NSU
NOVA SOUTHEASTERN UNIVERSITY

Analytical Coding: Performing Qualitative Data Analysis Based on Programming Principles

Abstract
In this paper, we argue that qualitative data analysis software lacks a tool that can be used to fulfill an algorithmic evaluation of conceptualization carried out in qualitative studies. We propose the context-oriented models of coding that conjugate single codes, that is, brief denotations made in natural language, by unusual local relationships called context-fixed elucidation (CFE). CFE is a local relationship between miscellaneous aspects of a case under study. The set of separate CFEs, originated by the analyst during conceptualization and called thesaurus, represents the case as a whole. On the basis of CFE structure and using the thesaurus' single codes as data, there is proposed an algorithm which calculates, without the involvement of the expert, whether there is or not global coherence of single codes used by analyst within the thesaurus. The tool thus obtained emulates for the codes originated in qualitative study the relationships known in the object-oriented programming, such as polymorphism, visibility, encapsulation, inheritance. A probe application of the new tool is demonstrated by the conceptualization of textual evidence. The application was performed with the help of a pilot computer package which architecture is based on the context-oriented models. Thanks to the models, QDAS can obtain special tools that would make researchers' analytical work more intelligible and coherent. The models proposed can find applications outside of research discourse including computer technologies used in various social spheres where people communicate in natural language.

Keywords
qualitative data, qualitative data analysis, context oriented methods, ontology, semantic networks

Acknowledgements
We would like to express our gratitude to Yuta Tamberg, who helped us with the English text for this article and whose patience and active concern cannot be overestimated.

The Qualitative Report 2021 Volume 26, Number 2, 316-333
https://doi.org/10.46743/2160-3715/2021.4342

Analytical Coding: Performing Qualitative Data Analysis Based on Programming Principles

Gennady Kanygin and Viktoria Koretckaia
Sociological Institute of the Russian Academy of Sciences, Branch of the Federal Center of Theoretical and Applied Sociology of the Russian Academy of Sciences, Saint Petersburg, Russia

In this paper, we argue that qualitative data analysis software lacks a tool that can be used to fulfill an algorithmic evaluation of conceptualization carried out in qualitative studies. We propose the context-oriented models of coding that conjugate single codes, that is, brief denotations made in natural language, by unusual local relationships called context-fixed elucidation (CFE). CFE is a local relationship between miscellaneous aspects of a case under study. The set of separate CFEs, originated by the analyst during conceptualization and called thesaurus, represents the case as a whole. On the basis of CFE structure and using the thesaurus' single codes as data, there is proposed an algorithm which calculates, without the involvement of the expert, whether there is or not global coherence of single codes used by analyst within the thesaurus. The tool thus obtained emulates for the codes originated in qualitative study the relationships known in the object-oriented programming, such as polymorphism, visibility, encapsulation, inheritance. A probe application of the new tool is demonstrated by the conceptualization of textual evidence. The application was performed with the help of a pilot computer package which architecture is based on the context-oriented models. Thanks to the models, QDAS can obtain special tools that would make researchers' analytical work more intelligible and coherent. The models proposed can find applications outside of research discourse including computer technologies used in various social spheres where people communicate in natural language.

Keywords: qualitative data, qualitative data analysis, context oriented methods, ontology, semantic networks

Introduction

The qualitative data analysis (QDA; Online QDA, 2018) is one of the promising trends in the development of methods for qualitative research. Contemporary QDA is carried out with the help of such computer packages as Atlas.ti, MAXQDA, NVivo, xSight, QDA Miner, Quirkos, Ethnograph and others, which constitute a separate class of applications called qualitative data analysis software (QDAS). One can easily find the description of any QDA package on the Internet.

Kelle (1997) considered the coding and retrieval functions as instrumental specifics of QDA software. Lewins and Silver (2007) saw the functionality of QDA packages as a set of tools - content searching, linking, coding, annotating, querying, mapping, or networking.

St. John and Johnson (2000) summarized the pros and contras of the packages, revealed by researchers applying this software in qualitative studies. During their 30-year development (Wolski, 2018), QDA packages were repeatedly subjected to comparative analysis of their instruments (Lewins & Silver, 2007; Pat Research, 2018; Tesch, 1994). After the KWALON (2016) conference, Evers (2018) outlined the directions of further development of QDAS. Among them there are: firstly, the elaboration of a QDAS package not dependent on the approach, which would facilitate researchers' teamwork. Secondly, addressing to the cloud architecture to make joint work more secure. Thirdly, comprehending to what extent "light" version of QDAS package can be useful.

To better understand the ways of possible development, we wish to draw particular attention to the idea of coding and its assumptions that underpins functionality of QDAS. We articulate this idea as traditional coding model (TCM). According to TCM, the coding is a process of semantic interaction of two native speakers (actors) - the informant and the researcher, who both describe the same event. TCM represents the informant as a witness (observer) of the event. The informant's description is, *par excellence*, a flow of text, called evidence or account.

By contrast, the researcher is not the observer of an event, but describes it through reconstructing the informant's evidence. Researcher makes the reconstruction by means of codes. The code is a short utterance in natural language (short textual denotation). It is an elementary analytic unit, constructed by the researcher in the process of reformulating the informant's evidence in a structural view instead of a textual stream. There are two reasons why this model treats codes as more thorough means of expressing the informant's meanings compared to the text flow. First, short formulations capture in a clearer fashion the informant's meanings within the analytic constructs of the researcher. In other words, the researcher's codes "atomize" the informant's opinion expressed by continuous textual flow. Second, by analyzing "atomized meanings" represented by a set of codes together, especially if linked (e.g., in a graph), the researcher gets a new instrument for revealing various uncertain connotations that are hard to catch in the textual flow.

The main methodological problem, solved instrumentally by TCM, is substantiation of the codes introduced by the researcher. Such substantiation is carried out through creating *primary codes* (i.e., analyst's own short textual designations) which he associates explicitly with a fragment or fragments of the informant's evidence. Since the informant is seen as a witness, the semantic interpretation of such a fragment serves as a "factual" verification of the code associated with the fragment.

To give the researcher a way to develop his own analytical suggestions, TCM assumes that, together with the primary codes, the researcher will create *secondary codes*. Secondary codes are also short verbal utterances formulated by the researcher according to his understanding of informant's account. However, within TCM, researcher should associate such codes not with fragments of informant's evidence, but with his own codes (both primary and secondary).

TCM sets a framework for coding, executed by the researcher during fieldwork. The main purpose of such coding is a clear structural representation of the witness's account. It seems clear, that to achieve this goal, the researcher has to introduce explicit links between primary and secondary codes.

Yet, TCM itself does not provide any methods for associating codes. This is why developers of QDA packages are compelled to complement the coding tool with instruments for codes binding, such as linking, queries, networking, etc. (see Lewins & Silver, 2007). However, these add-ons, developed outside TCM, do not supply QDA packages with the functionality researchers need. As La Pelle (2004) stated, QDA software abilities do not go beyond the basic operations with text, available to the users of an office program. Junker (2012) summed the situation: "They [QDA packages] allow in computerized form to do the same work that the previous generation of researchers performed with the help of cards and markers" (p. 5). As a result, Thompson (2002) noted that "the most difficult task for the researcher is the conceptual part of data analysis" (p. 16). Junker stressed the same common difficulty of coding, using a qualitative study by Franzosi, De Fazio, and Vicari (2012): "Their example also points to how much more complex analysis becomes once one moves beyond 'code and retrieve'" (as cited in Junker, 2012, pp. 85-86).

We emphasize the functional limitation of the coding model, implemented in existing QDA packages, since social scientists often recognize the coding as a core of the qualitative data analysis in general (Pierre & Jackson, 2014; Strauss, 1987). Likewise, the QDAS developers believe that coding and retrieval of qualitative data is the central function of their packages (MaxQDA, 2018; Quirkos, 2018).

Aiming to improve the QDAS tools in general, we offer an analytical coding model (ACM). In its computer implementation, the ACM provides the researcher with the methods developed in contemporary computer science - modular organization, encapsulation, visualization, compilation. Thus, operations with codes within QDA become actions performed by the researcher with tools that have proven effective in modern programming.

ACM consists of coding data structures and algorithms for their processing. This paper informally describes how ACM works while coding a textual evidence and allows the analyst to represent informant's account in the shape of algorithmically generated semantic networks. Aiming to demonstrate ACM as a functioning tool, we have implemented its data structures and algorithms as a part of our pilot computer ontoeditor. The papers (Kanygin, Poltinnikova, & Koretskaya, 2017; Poltinnikova & Kanygin, 2016) demonstrated earlier applications of the ontoeditor. To visualize the algorithm-generated results, together with ontoeditor, we also used Graphviz 2.38 (Graphviz, 2018).

Why analytical coding?

Working in many sociological projects, we faced three constantly repeating issues. First of all, the bigger data array we get during the project, the more complicated the idea of data comparing grows. A possibility of comparison becomes problematic because meanings go under in texts of research programs, informants' evidence, summaries and publications. In the sociological research, data are grounded in a conceptual framework constructed by researcher. Therefore, to make telling comparisons based on data one needs to have their explicit relationships to concepts researcher operates with when conceiving data collection process.

Secondly, if we are to analyze the qualitative data set in parallel with findings of other researchers, it is a complete disaster at the end to compare and align the results together in one summary document. The concepts, worded similarly in various studies, have desperately different meanings. The empirical studies convinced that sociological, especially, qualitative terminology is existentially polysemic and demands special means to handle permanent ambiguity.

Thirdly, in a repeating project, let us say a regular survey lasting five years long, it took us too much time to reveal facts and circumstances at various survey stages without repeating ourselves and multiplying our efforts already done. We started to think over how to keep the prior conceptual work and use it with necessary amendments but without redoing all over again.

As we were familiar with programming, we were stumbling upon analogies that looked promising to resolve our troubles. First of all, since merging data and theory takes place within researcher's intellectual reasoning, the researcher himself must demonstrate explicitly links between forwarded concepts and resulting data. However, because the society is "complex, rich and non-linear" (Urry, 2005, p. 235), there is a little chance to trace such links through publishing a volume written by a single scholar. Addressing ourselves to programming ideas, we have come across an idea that describing sophisticated social world the scholar should rely on teamwork performed by those whose every day personal interactions constitute the world itself.

We believe that a researcher who wants to coordinate his reasoning and data practically (i.e., under a working technology) should propose instrumental means that would allow any community of "informants" gathering in a team working together. Not "literary theory" (Allais, 1990), codebooks or hyperlinks known in qualitative data analysis, but, let us say, ontology-like methods aimed to describe any subjects or objects through explicit relationships among them. This is a way to connect theoretical ideas with data based on them as a whole conceptual framework being under permanent development by ordinary people (i.e., subjects of any social theory). The ontological methods (Chergui et al., 2020; Noy & McGuinness, 2000) seem to be a reliable tool to organize demanded teamwork.

In particular, as to our above-mentioned issues, thanks to polymorphism of ontological definitions researcher would be able to handle polysemy. Using inheritance chains would provide one with an effective way to eliminate redundant conceptual work. Compilation, an internal mechanism of ontologies, looks promising as well: it would permit to combine and at the same time to check for coherence the intellectual efforts of people having separate working tasks but gathered within a joint project.

However, to transform these analogies into computer instruments we have to overcome substantial trouble. Researcher cannot escape "story telling" (Charmaz, 2000) when conceptualizing informant's evidence. Neither can academician who writes a volume aimed to explain how society works. Therefore, the ontological methods as they are specified now (Unified Modeling Language [UML], 2020) look far-fetched to become researcher's instrument in sociology. The question is how to preserve every person's habit to describe anything in natural language and to put this habit under principles developed in science and programming?

In our below attempt to answer this question we bear in mind three points. Firstly, coding, as a scientific procedure, is the practice of intellectual interchanges between people, carried out by them through natural language. Secondly, we need an instrumental model that would permit a community of persons within one procedure, on the one hand, to apply their skills to describe whatever through habitual wordings and utterances, and on the other hand, to manage such speech manifestations with the help of ontology-like means. Thirdly, in order to imagine how the big universe works, we need to come up with differential equations that describe the oscillations of a small pendulum. Similarly in sociology, in order to understand how people become able to construct a big society based on little experience of each of them, we need to understand what the "differential equations" should be which describe the "pendulum" oscillating in mutual understanding between a researcher and his informant in the chamber case of their communication.

Structural mechanisms of analytical coding model

Just as QDAS coding tool, techniques based upon ACM aim to represent informant's evidence analytically. However, ACM provides instrumental innovations for many known qualitative data analysis techniques, including primary coding, code binding with a fragment of the informant's evidence, categorization of codes, and others. As a result, unlike TCM, ACM displays the outcome of coding in the form of an intuitive graph. To demonstrate what the ACM structural mechanisms are and how they work in applied conceptualization, we used a small textual example known as Terry text we found earlier at the site http://onlineqda.hud.ac.uk/:

> When you move into your own home, you're alone. There is no bustle of people around the house. I miss having someone to chat to when I get home. I put the TV or some music so there's some background noise, the silence makes me feel so alone. Sometimes I will be sat watching trash TV and thinking I should be out doing something rather than watching this rubbish. I read a lot but sometimes I am too tired and just want to veg out.

Paired denotation

Similarly to TCM, ACM allows the researcher to work with short verbal notations (codes), destined to re-formulate the informant's evidence. Unlike TCM, ACM requires the analyst to introduce such notations (called concepts or notions) during coding of any kind not separately, but in *pairs*. In ACM, the first element of the *pair* is called the *term* and the second – the *context*. All concepts, regardless to their position within the pair, constitute a set called the *dictionary*.

Formally speaking, the code pair is a novel component of ACM, not found in TCM. We propose the following informal interpretation of the ACM concept pair. The term is a traditional designation of an item or a matter, understood intuitively by a native speaker. The context indicates the condition, under which the term is used. Both the term and the context are concepts that a human analyst introduces, bearing in mind certain aspects of the situation he needs to describe. Thus, every paired denotation always presupposes a meaningful interpretation.

Now let us apply the paired denotation to the Terry's text. Initially, we introduce a concept TERRY EVIDENCE[1] to designate the text itself. Next, we make another new concept MY CONCEPTUALIZATION to indicate the conditions under which we consider Terry's account. Both concepts are our own and represent two acts of primary coding, well known in qualitative data analysis. Of course, within the former indication, we can see a real "object," (i.e., informant's text) and the latter indication describes something more versatile (i.e., the circumstances that relate to the text only through our own actions). However, we did not introduce anything besides these actions and so we have to state all the relevant circumstances, for carrying them out.

The situation above, where we have simultaneously introduced two concepts, makes us consider these concepts together. To document analytically the stated factual relationship between Terry's text and the circumstances in which we consider it, we join two separate concepts into a single pair <TERRY EVIDENCE, MY CONCEPTUALIZATION>.

Thus, we articulate explicitly that any analytic notation exists within some limitations, usually referred to as context. TCM also employs the idea of context to substantiate researcher's codes by binding them to fragments of the informant's evidence. However, within TCM a code and its context originate from two different actors: researcher (analyst) and informant (witness). We find it highly desirable for the researcher himself to establish the limitations for his own codes. If organized explicitly, such practice would put researcher's conceptual actions under instrumental control. The practice seems to be useful, since coding results *de facto* reflect not only informant's evidence, but also, implicitly, the analyst's own views. In addition, a multitude of the informants and variety of their opinions exacerbate the difficulties of sociological conceptualization.

Defining the context analytically does not mean that the researcher is allowed to ignore informants' opinions. Instead, thanks to ACM, he acquires control over the coherence of his own statements, substantiated by explicit links of any nature. ACM allows such control thanks

[1] In the text of the article, we use upper case to denote concepts that are analytical notations of ACM.

to its original algorithm: it constructs a semantic network from a set of paired denotations proposed by analyst to reconstruct informant's account. Depending on how successful and consistent the analyst was while inventing terms and coupling them with contexts, the resulting network would also become intuitively meaningful to any native speaker (see below).

We have explained the main features of paired notation with reference to the initial point of Terry's account conceptualization. Now let us continue conceptualizing this account according to the technique of qualitative data primary coding and with the help of ACM. We will use direct textual fragments from the evidence as concepts in order to simplify the presentation of paired notations during primary coding. In addition, we hope to avoid the undesirable modification of the informant's meanings, which often arises when researchers re-designate them as codes.

Consistent with the text of the informant, we enter the term WHEN YOU MOVE INTO YOUR OWN HOME. As the next step, required by the structure of ACM, we must specify the conditions under which we have created the term. Such conditions can be thought of as the real circumstances in which we carry out conceptualization. Above, we have already identified them with the help of the concept pair <TERRY EVIDENCE, MY CONCEPTUALIZATION>. Thus, before we ever introduced the concept WHEN YOU MOVE INTO YOUR OWN HOME, we have already attributed meanings to two concepts. To avoid ad hoc reasoning, we can use any of them as a context for a newly introduced term. Thus, remembering our earlier denotations, we make the pair <WHEN YOU MOVE INTO YOUR OWN HOME, TERRY EVIDENCE> that links the exact Terry's utterance to what we ourselves designated as his reflections. Naturally, this connection exists explicitly only in our analytical constructions.

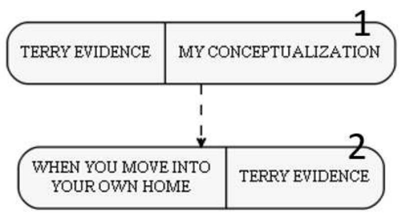
Continuing the primary coding, based on Terry's meanings, we introduce a next term YOU ARE ALONE. The account clearly suggests that YOU ARE ALONE occurred when Terry moved to his own home. Therefore, we can constitute another pair <YOU ARE ALONE, WHEN YOU MOVE INTO YOUR OWN HOME>. Following Terry's text, the next suitable term seems to be NO BUSTLE OF PEOPLE AROUND THE HOUSE, which conveys Terry's feelings, caused by his moving into a new home. In conjunction with this evidence and the logic of paired notation, we enter a new notion pair < NO BUSTLE OF PEOPLE AROUND THE HOUSE, WHEN YOU MOVE INTO YOUR OWN HOME>. As a result, WHEN YOU MOVE INTO YOUR OWN HOME becomes the context in which, according to the witness, we understand two terms YOU ARE ALONE and NO BUSTLE OF PEOPLE AROUND THE HOUSE.

As the coding under construction shows, the ACM assumes that any term can appear in the pair's body only in association with the context in which it is considered. First, we have introduced the notion TERRY EVIDENCE, and then we used it when specifying the context for the newly made concept WHEN YOU MOVE INTO YOUR OWN HOME. This already proposed concept served as a context for another new notion YOU ARE ALONE, and so on. A piece of conceptual work already accomplished is enough to demonstrate the main instrumental features of ACM. Up to now, we have created the dictionary that includes six concepts bound into four pairs. Below we shall enlarge the dictionary with new concepts, but now we are going to explain how analyst can relate concepts coupled in pairs.

Primary binding of concept pairs

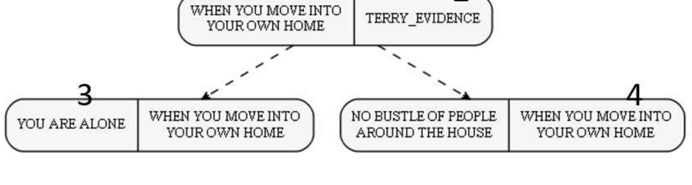
Linking, or binding, of analytical designations, both in ACM and traditional QDAS, is somewhat similar to making folders on a computer (e.g., /home/Documents/Library/Sociology/... etc. to an arbitrary depth): users can create limitless folders, following their own semantic considerations. The crucial difference of ACM, however, lies in the way pairs of concepts are associated to shape a single graph representing all concepts made during coding. ACM allows the user to create only two-level graphs with a single root vertex, as shown in Figures 1 and 2. These graphs are made of concept pairs and called *branchings*. A final, unified outcome of the analysis is called an *analytical coding results (ACR) graph*, or *semantic network,* appears thanks to the special ACM algorithm that generates it out of a complete set of separate branchings, supplied by analyst. We call this set of branchings a *thesaurus.* By contrast, within traditional QDAS researchers aggregate codes manually outside of the coding tool itself by devising graphs step-by-step. Being completely "hand-made" constructs, such graphs quickly become unwieldy and inoperable: their volume grows together with the evidence under study. Therefore, without adequate computer assistance to aggregate coding results, the researcher encounters troubles mentioned in the introduction.

Figure 1
Branching 1: initial branching opening the coding of Terry's evidence



In fact, analytical coding demands from the researcher some additional conceptual actions needed to make a thesaurus. We call this process *primary binding*. Upon completion, it provides several decisive benefits. Firstly, the human always creates a visual relationship between concept pairs, similar to folder-subfolder transition. Secondly, it becomes much easier for analyst to work with a net of big size, since the algorithm that compiles the net does not require human intervention. Thirdly, by comparing an evidence and a corresponding network informally, analyst can control which "semantic parts" of the evidence he incorporated into the network. The instrument to make such comparison gives the researcher a guide to create and modify concepts, pairs and their branchings until a generated semantic network assumes a desired shape and covers the informant's meanings seen by researcher in an evidence under study.

Figure 2
Branching 2: in the situation of Terry moving into his own home, we take into account two aspects of his residence: loneliness and the absence of sounds issued by other people.


2
WHEN YOU MOVE INTO YOUR OWN HOME
TERRY_EVIDENCE
3
YOU ARE ALONE
WHEN YOU MOVE INTO YOUR OWN HOME
4
NO BUSTLE OF PEOPLE AROUND THE HOUSE
WHEN YOU MOVE INTO YOUR OWN HOME

To illustrate this technique of linking concepts in ACM, we carry out a primary binding that would associate four concepts pairs described above in the framework of two branchings (see Figures 1 and 2). For convenience, the pairs within these branchings are numbered from 1 to 4. Proposing both branchings we rely "semantically" on the evidence. The branching 1 explains that our running interpretation (MY CONCEPTUALIZATION) of the Terry's text

(TERRY EVIDENCE) confines its semantics uniquely to the situation, which happens to the informant as he moves into his own home. The situation, titled WHEN YOU MOVE INTO YOUR OWN HOME, as well as our conceptualization, relies to the Terry's account, re-assigned above as TERRY EVIDENCE. The verb "confines" used above is one of many possible natural language interpretations of the relationship "one-to-many" or "parent-children" and brings no additional meaning to this intuitively understandable relationship.

In its turn, the branching 2 shows that, to our mind, Terry explains the situation of his new residence (already mentioned as the pair <WHEN YOU MOVE INTO YOUR OWN HOME, TERRY EVIDENCE>) through two aspects: by feeling of aloneness and by unexpected absence of sounds usually made by other people in the house. We can see the corresponding concepts YOU ARE ALONE and NO BUSTLE OF PEOPLE AROUND THE HOUSE in the child nodes of the branching 2. The verb "explains," just like "confines" above, is an optional natural language interpretation of "parent-children" relationship already expressed by the branching 2.

At the same time, the branching is a special structure that establishes the relationship "folder-subfolder" between pairs. Moreover, it can be used to instrumentally coordinate the researcher's operations with concepts during primary coding and primary linking. Let us remember, that by introducing a context within a pair we define the conditions of existence for a corresponding term. For instance, in the pair 2 (i.e., <WHEN YOU MOVE INTO YOUR OWN HOME, TERRY EVIDENCE>), we state that the term WHEN YOU MOVE INTO YOUR OWN HOME, to our mind, exists only within a specific case that we designated TERRY EVIDENCE. Thus, the first condition of a term existence in ACM, that we call *[pair] contextual* one, is the explicit indication of its context.

The second condition for the term's existence arises from the ACM linking of concepts within branchings by means of pairs. We can note that any child vertex in these constructs can exist only if a corresponding parent node was already created. Since any node in a branching is a concept pair, we propose the following rule for the node's existence: a *[branching] structural condition* (rule). This condition is exemplified in Figures 1 and 2: a child vertex (i.e., a concept pair within the branching) can exists only if the context of the child vertex pair is present in the parent node pair in either of two positions (i.e., as a term or as a context).

Thus, it turns out that the existence of the same term in the branching must satisfy the above two conditions. Both contextual and branching structural rules become possible due to ACM structural arrangement. However, when making branchings during primary binding researcher is not obliged to follow these rules. For instance, he may fill in nodes of these graphs with arbitrary concepts gathered by pairs. ACM accepts branchings with any structural features proposed by a human. The rules come into work only during ACM algorithms run. To see why it is so, let us consider their functioning in detail.

Algorithm for generating semantic network

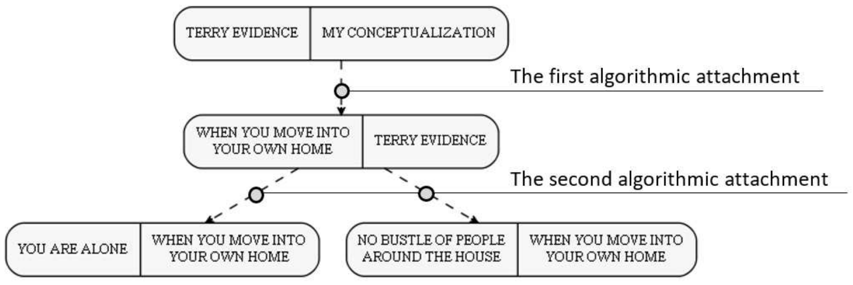
Here we report the basic algorithm that aggregates and manifests all concepts' connotations, which we had introduced as branchings, in a single ACR graph (i.e., in a unified semantic network). We demonstrate the functioning of the algorithm and its backgrounds with relation to our example. However, the algorithm can generate a network visualization of any thesaurus resulting from primary binding.

At the initial step, the algorithm demands to point out a starting concept pair. We choose the concept pair <TERRY EVIDENCE, MY CONCEPTUALIZATION>[2] from which we

[2] The same pair can enter both the thesaurus and the ACR graph. In the former case it is set in angle brackets, in the latter case we use figure brackets for its denotation.

started primary coding. This pair indicates uniquely in the thesaurus the first branching that we created to initiate our coding (Figure 1, branching 1). The algorithm makes the selected pair into a single root node {TERRY EVIDENCE, MY CONCEPTUALIZATION} of the semantic network being formed. Further, it considers all nested pairs, given by the branching 1, located in thesaurus. Because of our primary linking, in the thesaurus there is only one such pair - <WHEN YOU MOVE INTO YOUR OWN HOME, TERRY EVIDENCE> (see Figure 1, pair 2). After having found this pair, the algorithm checks if it satisfies the branching structural condition but with reference not to parent node in the branching 1 but to the parent vertex already made in the graph under construction. At this stage the graph contains a single root vertex {TERRY EVIDENCE, MY CONCEPTUALIZATION}. Since the notion TERRY EVIDENCE enters the single root node and the pair under joining <WHEN YOU MOVE INTO YOUR OWN HOME, TERRY EVIDENCE> obviously includes this notion as context, the structural rule is satisfied. Therefore, the joining pair becomes a new node under the root vertex (see Figure 3, the first algorithmic attachment).

Figure 3
Algorithmic binding of branchings: The simplest case of generating a semantic network

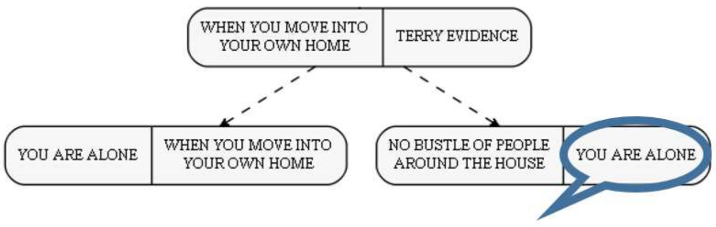

TERRY EVIDENCE
MY CONCEPTUALIZATION
The first algorithmic attachment
WHEN YOU MOVE INTO YOUR OWN HOME
TERRY EVIDENCE
The second algorithmic attachment
YOU ARE ALONE
WHEN YOU MOVE INTO YOUR OWN HOME
NO BUSTLE OF PEOPLE AROUND THE HOUSE
WHEN YOU MOVE INTO YOUR OWN HOME

After having attached the second node to ACR graph, the algorithm scans the thesaurus through all branchings to find those in which the just attached pair {WHEN YOU MOVE INTO YOUR OWN HOME, TERRY EVIDENCE} constitutes the branching root vertex. There is only one branching (Figure 2, branching 2) that meets this criterion. Branching 2 has two nested pairs (<YOU ARE ALONE, WHEN YOU MOVE INTO YOUR OWN HOME> and <NO BUSTLE OF PEOPLE AROUND THE HOUSE, WHEN YOU MOVE INTO YOUR OWN HOME >) each of which will become a new node of the ACR graph. However, just before executing this possible attachment the algorithm again checks the branching structural rule to verify whether any of the pairs found corresponds to the parent node {WHEN YOU MOVE INTO YOUR OWN HOME, TERRY EVIDENCE} already existing in the graph. Figure 3 shows, that the branching structural rule is fulfilled at the second algorithmic attachment as well.

Now the algorithm has made two new nodes: {YOU ARE ALONE, WHEN YOU MOVE INTO YOUR OWN HOME} and {NO BUSTLE OF PEOPLE AROUND THE HOUSE, WHEN YOU MOVE INTO YOUR OWN HOME}. The next step is to resolve whether these nodes can appropriate their child vertexes. To do so, the algorithm looks in the thesaurus and decides if there is a branching that contains each of pairs (<YOU ARE ALONE, WHEN YOU MOVE INTO YOUR OWN HOME> and <NO BUSTLE OF PEOPLE AROUND THE HOUSE, WHEN YOU MOVE INTO YOUR OWN HOME >) at the root position. Since there is no branching like that in our primary binding, the algorithm stops. The resulting ACR graph (Figure 3) is limited to four nodes. So far, we have demonstrated how the algorithm can generate a common graph out of disjunct results, made by researcher through primary binding.

Figure 4

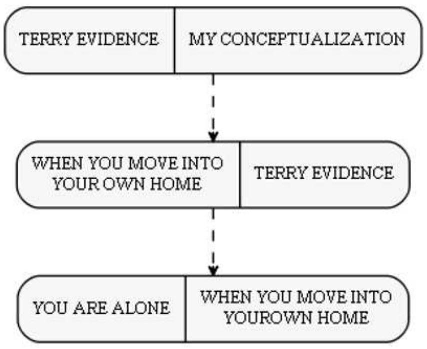
Branching 3: One with a pair of concepts, which context YOU ARE ALONE does not enter in the parent pair at either of two positions


WHEN YOU MOVE INTO YOUR OWN HOME
TERRY EVIDENCE
YOU ARE ALONE
WHEN YOU MOVE INTO YOUR OWN HOME
NO BUSTLE OF PEOPLE AROUND THE HOUSE
YOU ARE ALONE

Of course, from the very beginning of our analytical work we have kept in mind contextual and structural rules that underpin ACR graphs. In addition, the volume of our former analytical work is so tiny, that we had no difficulty in following these rules. Nevertheless, we could have made a mistake, e.g., when binding concepts in pairs and, as stated above, the ACM would accept our mistake. For example, if we convert the branching shown in Figure 2 into the branching of Figure 4 (in the pair 4 of branching 2 the context WHEN YOU MOVE INTO YOUR OWN HOME is replaced with the context YOU ARE ALONE), the resulting graph will assume the structure shown in the Figure 5. The pair <NO BUSTLE OF PEOPLE AROUND THE HOUSE, YOU ARE ALONE> will not be part of the semantic network, because context YOU ARE ALONE does not enter the parent pair at either of its two positions for concept.

Figure 5

The algorithmic binding of branchings: the pair <NO BUSTLE OF PEOPLE AROUND THE HOUSE, YOU ARE ALONE> does not enter in the generated network due to inappropriate context YOU ARE ALONE.

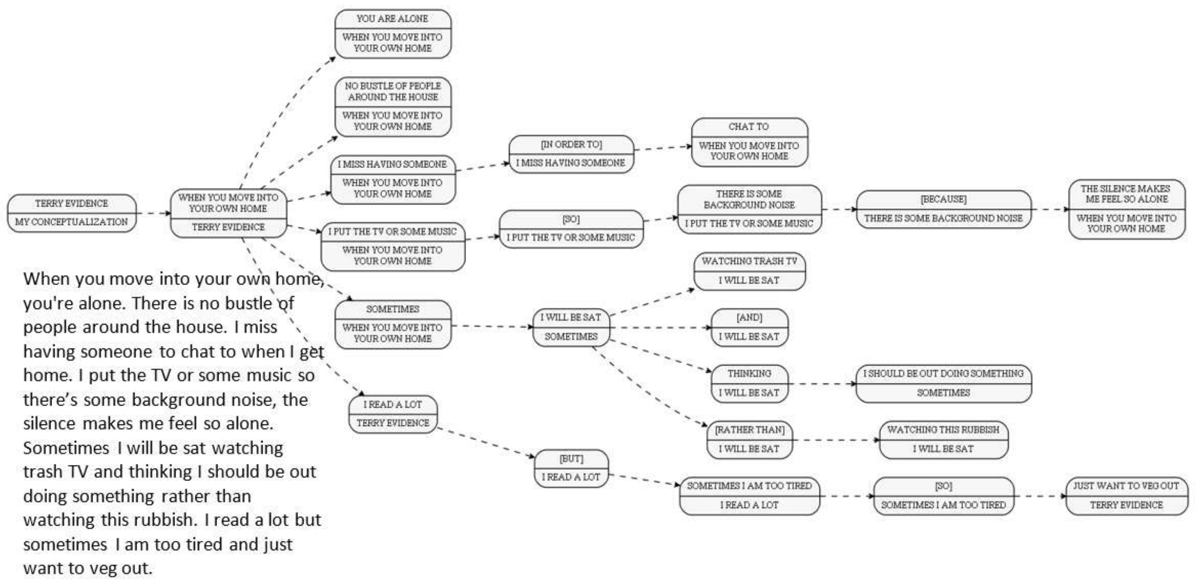

TERRY EVIDENCE
MY CONCEPTUALIZATION
WHEN YOU MOVE INTO YOUR OWN HOME
TERRY EVIDENCE
YOU ARE ALONE
WHEN YOU MOVE INTO YOUROWN HOME

In general, if the user deviates from either of the formulated rules, it will affect the shape of the generated semantic network. We find the main deviation to be a mismatch between the term and its context. However, we cannot exclude other inconveniences in analyst's work that will be manifested by the ACR graph.

The controlling facilities of the ACM algorithm become more apparent, if we expand the above rules for the existence of concepts within ACM. Such extension becomes possible because of new instrumental acquisitions due to the algorithmic construction of network. In contrast to a human, the algorithm operates equally well on graphs of all sizes. It can work not only with the branchings (i.e., structurally restricted graphs), but with the entire ACR graphs, generated by the algorithm itself.

There are many ways to convert this consideration into algorithmic rules. For instance, instead of checking whether the context of a given nested pair is present in the parent vertex (at term or context position, i.e., applying the branching structural rule), the algorithm can look through the whole branch where the nested pair is about to become a new leaf node. If there is at least one such vertex, the graph under construction acquires a new leaf node with the given nested pair. Let us call this new convention the *extended structural rule*.

Figure 6
Semantic network constructed for Terry's evidence in man-computer manner by means of analytical cording algorithms


TERRY EVIDENCE
MY CONCEPTUALIZATION
WHEN YOU MOVE INTO YOUR OWN HOME
TERRY EVIDENCE
YOU ARE ALONE
WHEN YOU MOVE INTO YOUR OWN HOME
NO BUSTLE OF PEOPLE AROUND THE HOUSE
WHEN YOU MOVE INTO YOUR OWN HOME
I MISS HAVING SOMEONE
WHEN YOU MOVE INTO YOUR OWN HOME
[IN ORDER TO]
I MISS HAVING SOMEONE
CHAT TO
WHEN YOU MOVE INTO YOUR OWN HOME
I PUT THE TV OR SOME MUSIC
WHEN YOU MOVE INTO YOUR OWN HOME
[SO]
I PUT THE TV OR SOME MUSIC
THERE IS SOME BACKGROUND NOISE
I PUT THE TV OR SOME MUSIC
[BECAUSE]
THERE IS SOME BACKGROUND NOISE
THE SILENCE MAKES ME FEEL SO ALONE
WHEN YOU MOVE INTO YOUR OWN HOME
SOMETIMES
WHEN YOU MOVE INTO YOUR OWN HOME
I WILL BE SAT
SOMETIMES
WATCHING TRASH TV
I WILL BE SAT
[AND]
I WILL BE SAT
THINKING
I WILL BE SAT
I SHOULD BE OUT DOING SOMETHING
SOMETIMES
[RATHER THAN]
I WILL BE SAT
WATCHING THIS RUBBISH
I WILL BE SAT
I READ A LOT
TERRY EVIDENCE
[BUT]
I READ A LOT
SOMETIMES I AM TOO TIRED
I READ A LOT
[SO]
SOMETIMES I AM TOO TIRED
JUST WANT TO VEG OUT
TERRY EVIDENCE

When you move into your own home, you're alone. There is no bustle of people around the house. I miss having someone to chat to when I get home. I put the TV or some music so there's some background noise, the silence makes me feel so alone. Sometimes I will be sat watching trash TV and thinking I should be out doing something rather than watching this rubbish. I read a lot but sometimes I am too tired and just want to veg out.

To demonstrate how this wider rule works, we extend our conceptualization of Terry's text by adding more branchings to the existing thesaurus. Using the paired coding and binding techniques described above, we have reformulated the textual fragment shown at the bottom left of the Figure 6. The complete thesaurus, its extended dictionary and text for analysis are present at the links http://coknowledge.ru/wp-content/uploads/2018/11/TerryThesaurus.txt http://coknowledge.ru/wp-content/uploads/2018/11/TerryDictionary.txt http://coknowledge.ru/wp-content/uploads/2020/10/TerryText.txt. The former dictionary and thesaurus of two branchings have become incorporated into their newly created counterparts.

When constructing this extended thesaurus, we have departed slightly from using only informant's words and added our own notation along with the formulations of the informant. This conceptual deviation consists of "auxiliary words" that are absent in the evidence, for instance IN ORDER TO, and BECAUSE. In this way, we tried to smoothen the transitions between the situations, which we encounter in Terry's text and represent on the graph by means of concepts. We combine our "auxiliary words" with Terry's analogs, such as SO, BUT, etc. We use square brackets to show auxiliary words within the thesaurus.

The renewed thesaurus allows us to generate a new semantic network (see Figure 6). The working of the extended structural rule is best illustrated by the node {JUST WANT TO VEG OUT, TERRY EVIDENCE}. The node demonstrates that the term JUST WANT TO VEG OUT is defined in the context TERRY EVIDENCE. However, this context enters not in the parent node but in the vertex {I READ A LOT, TERRY EVIDENCE}, which is located higher along the graph's branch on which the {JUST WANT TO VEG OUT, TERRY EVIDENCE} is the leaf (see Figure 6). The extended structural rule works in the similar fashion, e.g., for the terms CHAT TO and THE SILENCE MAKES ME FEEL SO ALONE.

In the framework of ACM, the extended structural rule formalizes in the framework of ACM the idea of restricting arbitrary statements during coding work executed by analyst. The algorithm regulates the analytical work by prescribing the order in which the analyst should enter concepts and bind them in branching. This order reflects not the external circumstances, for example, arrangement of conceptual actions in time, but the very semantics of the evidence subjected to analytical reformulation. Thus, the generated network is simultaneously a tool for regulation of, and assistance to, primary binding executed by researcher.

At every step of construction, the semantic net suggests to the analyst the contexts in which new concepts may be introduced. For example, one may extend the branch {TERRY EVIDENCE, MY CONCEPTUALIZATION}, {WHEN YOU MOVE INTO YOUR OWN HOME, TERRY EVIDENCE}, {YOU ARE ALONE, WHEN YOU MOVE INTO YOUR OWN HOME}. To do so, we can make a new child node, say {NEW NOTION, ITS CONTEXT} and place it under the parent {YOU ARE ALONE, WHEN YOU MOVE INTO YOUR OWN HOME}. To see the result in a generated network, we need to substitute ITS CONTEXT with any concept, already present in the vertexes of the branch. Such visual regulation of choosing contexts for newly introduced terms substantially facilitates analyst's efforts of elaborating branchings.

Our examples and their interpretation demonstrate that the algorithmic representation of the paired coding and linking relates only to the analytical actions of the researcher. The algorithms do not answer the question whether the informant's evidence has any link to the reality and, if so, to what extent. The algorithms can only verify the internal coherence of paired binding through demonstrating the network, without being able to connect the observable denotations with the outside world. Of course, such verification becomes possible if one assumes ACM as means to describe a sociological plot.

Semantic network as the result of analytical coding

Instrumentally speaking, the ACM allows building up a graph that visualizes the results of the analytical coding. Keeping in mind the goal of coding in QDA, we also refer to this graph as a semantic network that represents informant's meanings.

Interpretation of ACR graph

As the figures demonstrate, ACM algorithms generate the ACR graph in the form of a tree composed of pairs of concepts. In it, one can trace the chains of reasoning taken from the text of the informant. When reading these chains, we keep in mind our initial interpretation of the notation by means of pairs. The first concept in a node (term) is a traditional primary code. The second concept (context) designates the conditions or domain of existence of the term. Our appropriation of concepts as terms or contexts follows informant's words.

In terms of QDAS, our semantic network arises from the primary coding, based on the verbal designations of the informant himself. However, there is a fundamental difference: the network demonstrates the areas of codes' existence with the help of the codes themselves. This difference is predicated on the functioning of a fundamentally new mechanism for linking analytical notations made in the form of natural language utterances. Such mechanism is absent in traditional QDAS.

Various conceptualizations of the same evidence

As well as QDAS, ACM does not automatically reconcile evidently incompatible coding results, obtained from the same evidence by different people and, even by the same person working in various circumstances (Bazeley, 2012). Certainly, the ACR graph represents one of many possible conceptualizations or explicit interpretations, coming from Terry's text. Each of them is based on the author’s assumptions that underpin every conceptual picture. Assumptions seem to be key point of any interpretation and therefore should be included in it explicitly. However, this recommendation is hard to follow in practice (see Introduction).

For example, our first assumption was the task we wanted to solve through analytical reasoning. We could formulate this task as covering the meanings one can see in Terry’s text. Although we constructed a corresponding ACR graph, there is no explicit indication of the very task, which called the graph into being. In addition, one of our assumptions was the use of the informant’s own utterances as analytical notation. However, the ACR graph does not contain any explanation for our decision. Another assumption allowed us to use auxiliary words in the hope to smoothen the interpretation of semantic network obtained. Again, the semantic net misses any mention of why we have used words in square brackets. Explanation and elaboration of the assumptions is entirely up to the analyst. Below we will discuss how ACM approach can help the analyst to manifest these assumptions.

Discussion

Qualitative data analysis and analytical coding

Coding is the key operation, which had given rise to the QDA software, and ACM is a new structural model, that can change the traditional view of this procedure. However, the researcher who uses a QDA package is unlikely to be interested directly in the models on which the package is built. The purpose of the researcher is to resolve his own methodological difficulties by means of a package. One of such difficulties is the problem of variability of interpretations of the same text. Bazeley describes the problem in this way: "Being interpretive means that coding is therefore inherently subject to variability of interpretation by different researchers, and even, perhaps, by the same researcher at different times" (Bazeley, 2012, p. 77).

This statement allows us to see at least two "conceptual needs" that the researcher feels when coding, and the QDA tools must satisfy. The first need is to ensure that the coding results correctly represent his own interpretation of the informant's account. This interpretation cannot be final, but it must be "structurally articulated" (i.e., be expressed by a distinct structural model) for example, by a semantic network. This structural certainty of the model, as opposed to the informant's vague flow, is a condition *sine qua non* for further development of the researcher's own version and for comparisons between other interpretations. However, due to the inherent variability of even one interpretation, the structural coding model must be easily scalable to enable the researcher to follow his own rapidly changing views with explicit model designations. The nature of researcher-informant fieldwork communication is such, that the researcher faces unlimited volume of "semantics" that constitute sociological plot. Hence, any structural model of the coding should also be operable independently of the volume of evidence (the number of informants, the vastness of the plot, etc.).

The second need of the researcher is the management of the multiple interpretations of the same evidence. As Bazeley notes, the authors of such interpretations can be either different researchers, or the same person, who creates conceptual versions under different circumstances. In programming, we can see a similar situation of "conceptual discord": to solve the same task various programmers will write programs that differ conceptually and textually. If forced to merge such programs, the programmers will apply a special teamwork tool. It will not automatically reconcile and put together existing programs, but rather it will facilitate collective work of many programmers. To create such a teamwork instrument, the analytical efforts of each programmer should be subjected to the same known principles - modularity, encapsulation, visualization, etc.

We propose the similar approach to the problem of variability of interpretations during coding. As a first step in this direction, ACM allows the researcher to organize his coding work on the aforementioned programming principles, thus giving basic structural clarity to a separate interpretation of the informant's evidence.

ACM facilities to code and retrieve qualitative data
Modularity of ACM

Any vertex of a semantic network contains two concepts: the second one plays the role of a module, or a block, and thus ensures the existence of the first one. Note that within ACM the same concept can play both roles: a “variable,” which is defined with respect to the “module,” and a “module” with respect to which some other "variable" is defined. For example, in one node the concept WHEN YOU MOVE INTO YOUR OWN HOME is the term, and in another the same concept is the context (see Figure 6).

Within ACM framework, all conceptual actions of the user during analytical coding take form of the structurally separate components: concepts, pairs and branchings. It is important that, as our examples show, each such component has a clear semantic interpretation. Interpretation of the semantic network is based on the meanings that were laid by the analyst in the primary coding and binding. How the semantically meaningful, but structurally disconnected coding results are assembled in a single whole is shown in detail in the text of the article above.

However, to see the expected coding results, the user has to understand and take into account how ACM works during primary coding and binding. TCM organization of the codes does not demand the analyst to supply his own denotations that mark the boundaries of codes' existence. By contrast, ACM offers the analyst to indicate, by means of codes, the boundaries of the existence of codes themselves. This offer brings together the work of a researcher with the work of a programmer.

Visual representation of the ACM modularity

All analytical constructions, created by researcher on the base of ACM (i.e., pairs and branchings) are graphs. In this way, ACM provides visual representation of all analytical actions of the researcher. The pair, composed of concepts (i.e., natural language utterances), looks no less clear than the traditional code or memo. The branching is a fully intelligible two-level graph, composed of concept pairs. Finally, a semantic network is a connected graph of any size, algorithmically generated from concept pairs. ACM does not use any means of linking concepts other than observable graphs. These graphs only associate to one another meaningful utterances in the natural language.

Compilation of a unified ACR graph

The ACR graph is a unified structure of knowledge, certain aspects of which are communicated by informants and are analytically recorded by a researcher. Formally, sociological knowledge, obtained with the help of ACM, is algorithmically compiled in the form of a graph of a special type based on a set of individual branchings. Informally, compilation is a powerful mechanism that makes operability of ACR independent of the volume of evidence under study.

Scalability of ACR graph

The modular and visual arrangement, as well as compiling algorithms predefine versatile scalability of an ACR graph. While introducing the notions in order to conceptualize Terry's text, we borrowed his formulations. Any native speaker can further expand each of these formulations based on human understanding of the informant's words. For example, one can clarify primary code WHEN YOU MOVE INTO YOUR OWN HOME by bringing and connoting the concepts WHEN, MOVE, HOME and others at his discretion. Such structural add-ons demands from an analyst nothing more than a general human ability to understand what "home," "when," etc. mean. Instrumentally, the analyst should execute primary coding and binding described above. It is very encouraging that to modify semantic network of any magnitude the analyst should create a number of individual branchings, while the complete semantic view will be compiled algorithmically. Done in this way, all conceptual modifications are local, they do not depend on one another and happen under visual control of the analyst.

Shallow learning curve

To apply ACM tools in practical conceptualization, the researcher does not have to understand how the ACM algorithms work. Instead, he should acquire certain practical skills to handle concepts and their visually presented relationships (pairs, branchings, and networks). Similarly, a computer user may not know what are the algorithms behind a text editor, but such ignorance does not prevent him from using it successfully to work with texts.

Our experience with ontoeditor shows that the prejudice against analytical methods, especially in the field of qualitative sociology, as well as the novelty of paired denotation and compilation techniques, initially lead to many questions. However, practical work quickly provides answers to these questions and users easily assimilate analytical coding methods. Indeed, ACM imposes very few requirements on the competence of the analyst. Firstly, he must know the natural language in order to be able to enter concepts. Secondly, he should understand why and how to create a "folder-subfolder" construct – a relationship, widely used in computer applications. Mastery of this relationship is needed to connect concepts with each other. Clearly, these requirements are easily met. This, combined with exclusively visual nature of the tools used in introducing and binding concepts, makes training in analytical coding methods a quick and uncomplicated task.

ACM resistance to analyst's arbitrary conceptualizing

Since all concepts and their relationships arise from human actions, ACM imposes a general rule, which the conceptual work of the researcher obeys. The rule is pragmatic in nature: the researcher may create his own analytical notations – concepts – only after he has specified the conditions for their existence. The necessity of defining a concept before it can be used lies at the heart of all analytical methods. For example, in computer science: before a programmer can use a variable in an operator, he must identify it relative to a corresponding block.

ACM enforces, rather than recommends, the contextual certainty of concepts. ACM algorithms generate the semantic network demonstrating visually whether the analyst has preserved the contextual certainty of his designations. In order for the algorithm to work, the researcher has to perform not only the primary coding, but also the primary binding of concept pairs he proposed. However, this additional work is justified: the ACM places the arbitrary utterances made by the researcher under instrumental control. Human coder becomes "conceptually obligated" to take into account what he has already uttered by means of notation.

It is important that such a "conceptual obligation" arises not because of an alien will, but because of his own analytical suggestions made earlier.

The result of conceptualization of the informant's text using ACM tools is a compiled semantic network. Considered together, the source text and the semantic network, allow any native speaker to judge the quality of the analytical reproduction of the informant's meanings, done by the researcher. The reader can measure our analytical success in respect of Terry's text.

Conclusion

We argue that ACM is an original structural mechanism capable to expand significantly the functionality of QDAS and facilitate the analytical work of researchers through its full-fledged software implementation. A new functionality can promote to solve such issues as: disclosing the ambiguity of verbal formulations, tracing their dependencies on the context, reconciling the meanings of the statements proposed by different people, checkup of the consistency of researcher's analytic statements, visualization through graphs of semantic connotations hidden in informants' evidence. Such means have proved all their effectiveness in programming and would facilitate arranging sociological discourse through prototyping, polymorphism, encapsulation, teamwork and other constructive techniques.

To assess the broader prospects of our findings, one should notice that ACM represents an original way of operating knowledge in two dissimilar approaches at once. Let us represent them in "social faces." A field researcher coordinates practically his meaningful actions with other people through natural language. A Hi-Tech developer shares his intellectual practice with colleagues through ontological methods. The ACM merges both kinds of intellectual interactions. Such merging of informal and formal methods of knowledge management in one instrumental procedure seems to be promising as to unifying knowledge formats used by humans and computer agents in modern technologies of digital society (Dragicevic et al., 2020).

References

Allais, M. (1990). La science économique d'aujourd'hui et les faits. *Revue des Deux Mondes, 4*, 54–74.

Bazeley, P. (2012). Regulating qualitative coding using QDAS? *Sociological Methodology, 42*(1), 77-78.

Charmaz, K. (2000). Grounded theory: Objectivist and constructive methods. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (2nd ed., 509-535). Sage.

Chergui, W., Zidat, S., & Marir, F. (2020). An approach to the acquisition of tacit knowledge based on an ontological model. *Journal of King Saud University – Computer and Information Sciences, 32*(7), 818-828. https://doi.org/10.1016/j.jksuci.2018.09.012

Dragicevic, N., Ullrich, A., Tsui, E., & Gronau, N. (2020). A conceptual model of knowledge dynamics in the industry 4.0 smart grid scenario. *Knowledge Management Research & Practice, 18*(2), 199-213.

Evers, J. C. (2018). Current issues in qualitative data analysis software (QDAS): A user and developer perspective. *The Qualitative Report, 23*(13), 61-74. http://nsuworks.nova.edu/tqr/vol23/iss13/5

Graphviz. (2018). *Graphviz- Graph visualization software*. Retrieved from https://graphviz.gitlab.io/download/

Junker, A. (2012). Optimism and caution regarding new tools for analyzing qualitative data. *Sociological Methodology, 42*(1), 85-87.

Kanygin, G., Poltinnikova, M., & Koretskaya, V. (2017). Experience of building social knowledge based on computer ontological methods. *Sociological Journal, 23*(3), 25–41 (in Russian).

Kelle, U. (1997). Theory building in qualitative research and computer programs for the management of textual data. *Sociological Research Online, 2*(2). http://socresonline.org.uk/2/2/1.html

KWALON. (2016). *KWALON Conference 2016: Reflecting on the future of QDA software: Chances and challenges*. Retrieved 20.10.2020 from https://www.tijdschriftkwalon.nl/inhoud?jaar=21&nummer=1

La Pelle, N. (2004). Simplifying qualitative data analysis using general purpose software tools. *Field Methods, 16*(1), 85-108.

Lewins, A., & Silver, C. (2007). *Using qualitative software: A step-by-step guide*. Sage.

MaxQDA. (2018). Retrieved 20.10.2020 from https://www.maxqda.com/qualitative-data-analysis-software

Noy , N. F., & McGuinness, D. L. (2000). *Ontology development 101: A guide to creating your first ontology*. Knowledge Systems Laboratory. http://ftp.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf

Online QDA. (2018). Retrieved 20.10.2020 from http://onlineqda.hud.ac.uk/

Pierre, E. St., & Jackson, A. Y. (2014). Qualitative data analysis after coding. *Qualitative Inquiry, 20*(6), 715–719.

Quirkos. (2018). Retrieved 20.10.2020 from https://www.quirkos.com/learn-qualitative/qualitative-analysis-software.html

Poltinnikova, M., & Kanygin, G. (2016). Graphic ontological methods to compile collective knowledge of social processes. *Proceedings EGOSE 2016: Electronic Governance and Open Society: Challenges in Eurasia - 2016* (pp. 223-228).

St. John, W., & Johnson, P. (2000). The pros and cons of data analysis software for qualitative research. *Journal of Nursing Scholarship, 32*(4), 393–397.

Strauss A. L. (1987). *Qualitative analysis for social scientists.* Cambridge University Press.

Tesch, R. (1990). *Qualitative research: Analysis types and software tools*. The Falmer Press.

Thompson, R. (2002). Reporting the results of computer-assisted analysis of qualitative research data [42 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* [On-line Journal], May, 3(2). http://www.qualitative-research.net/index.php/fqs/article/view/864/1878

Pat Research. (n.d.). *Top 16 qualitative data analysis software*. Retrieved 20.10.2020 from https://www.predictiveanalyticstoday.com/top-qualitative-data-analysis-software/

UML. (2020). *Unified Modeling Language. About the Unified Modeling Language Specification Version 2.5*. https://www.omg.org/spec/UML/2.5/About-UML

Urry, J. (2005). The complexities of the global. *Theory, Culture & Society, 22*(5), 235-254.

Wolski, U. (2018). The history of the development and propagation of QDA software. *The Qualitative Report, 23*(13), 6-20. http://nsuworks.nova.edu/tqr/vol23/iss13/2

Author Note

Gennady Kanygin is the Leading Researcher at Sociological Institute of Russian Academy of Sciences - the branch of Federal Center of Theoretical and Applied Sociology of the Russian Academy of Sciences in Saint Petersburg. His methodological interests include computer methods in sociology, computer-aided interviewing, qualitative data analysis, and knowledge management. He has published in each of these areas. Please direct correspondence to g.kanygin@gmail.com.

Viktoria Koretckaia is a Researcher at Sociological Institute of Russian Academy of Sciences - the branch of Federal Center of Theoretical and Applied Sociology of the Russian Academy of Sciences in St. Petersburg. She participates in several science and commercial research and development projects related to knowledge management systems. Her methodological interests include computer methods in sociology, knowledge management, databases' structure engineering, machine learning, information systems engineering. She has published in Russian in the area of Knowledge Management. Please direct correspondence to interperfection@gmail.com.

Acknowledgements: We would like to express our gratitude to Yuta Tamberg, who helped us with the English text for this article and whose patience and active concern cannot be overestimated.

Article Citation

Kanygin, G., & Koretckaia, V. (2021). Analytical coding: Performing qualitative data analysis based on programming principles. *The Qualitative Report, 26*(2), 316-333. https://doi.org/10.46743/2160-3715/2021.4342