

Systematic Versus Interpretive Analysis with Two CAQDAS Packages: NVivo and MAXQDA

Elif Kuş Saillard

Key words:
grounded theory
methodology;
CAQDAS; NVivo;
MAXQDA

Abstract: The purpose of this study is to compare two different Computer Assisted Qualitative Data Analysis (CAQDAS) packages (NVivo and MAXQDA) on a specific aspect. The same data from an auto-photographic study is analyzed using the same approach to make the comparison. The comparison is not based on a data oriented evaluation, but the methodological approach used constitutes the basis for the evaluation. The criteria for the evaluation of these tools are: closeness to the data, ease of coding and memoing, and *the interrelationship among the data, code and the memo* which were derived from the methodological approach. The first level coding process was accomplished using Grounded Theory Methodology (GTM) with two different CAQDAS packages. As for the result, in the GTM interpretation is crucial and MAXQDA supports the interpretive style better than NVivo.

Table of Contents

- [1. Introduction](#)
- [2. Reflections on Methodology](#)
 - [2.1 My data](#)
- [3. First-Level Coding Process with Two CAQDAS Packages](#)
 - [3.1 Terminology](#)
 - [3.2 Interfaces: Coding workspaces](#)
 - [3.3 Open coding and in-vivo coding](#)
 - [3.4 Coding Tool Bar](#)
 - [3.5 Adding reflexive notes: Memos, comments, annotations](#)
- [4. Reorganizing the Codes: Checking Memos, Annotations, Comments](#)
- [5. Conclusion](#)

[References](#)

[Author](#)

[Citation](#)

1. Introduction

Computer Assisted Qualitative Data Analysis (CAQDAS) packages have a thirty year history. CAQDAS refers to software packages that are developed especially to support *qualitative approaches* to qualitative data (LEWINS & SILVER, 2007). There are more than 20 packages available on the market today. Some of the CAQDAS packages popular today are (in alphabetical order): ATLAS.ti, HyperRESEARCH, MAXQDA, NVivo and QUALRUS. [1]

As expressed by Nigel FIELDING at the CAQDAS 07 Conference in 2007:
"CAQDAS choice is like buying a car—there is no such thing as the one 'best car' but rather the 'right car' for your mix of needs and purposes" (FIELDING & LEE,

2007, p.5). CAQDAS packages that are in the same category, such as theory-building, are equivalent concerning the sophistication of tools that support various stages of analytical process. Besides, these packages do not impose any methodological approach by allowing the researcher to apply different analysis strategies (inductive or deductive, interpretative or systematic). [2]

Still, the answer to the most common question "which is the best CAQDAS package?" is not easy. There are studies comparing CAQDAS packages in a more broad sense, especially Ann LEWINS and Christina SILVER's "Using Software in Qualitative Research" (2007) which is an inclusive and very useful book. However more user-based evaluations of CAQDAS packages should be undertaken to make clear how differences in "software architecture" differentiates the way various packages supports analysis. The tools offered by CAQDAS packages "may support certain tasks differently and there is some debate as to whether a particular package may 'steer' the way you do the analysis" (p.8). Thus, I use the term "software architecture" to refer to how the tools are designed and how they are accessed. Even though the theory-building software packages offer functions and tools that support the methodological variety in qualitative studies, software architecture may have an implicit effect on your way of doing analysis. To be aware of this effect, the analyst should really be conscious about what s/he needs and this consciousness can only originate from a methodological reflexivity. [3]

The evaluation or comparison of these software packages is significant in terms of methodological preferences. Therefore, in this study my comparison of two popular CAQDAS packages, MAXQDA 10 and NVivo 8 is based on Grounded Theory Methodology (GTM) (CHARMAZ, 1983, 2006) as methodological preference. My comparison focuses only on the first level coding process which is for me the core of the GTM and which combines both interpretative and systematic approaches. Accordingly, I focus on the tools that are functional in this process and I make my evaluations based on three criteria: 1. closeness to the data, 2. ease of coding and memoing, 3. the interrelationship among the data, code and memo. [4]

2. Reflections on Methodology

Qualitative methodology covers a wide spectrum of approaches ranging from post-positivism to postmodernism. The chosen approach will affect the research design and the analysis strategy. For example, analysis strategy is classified into two main different styles: systematic (indexing, coding approach) or interpretative (cross-referencing, hypertext linking) (KELLE, 1997). [5]

Grounded theory methods are considered to support a "systematic" approach to the analysis because "coding" is pivotal (COFFEY, HOLBROOK & ATKINSON, 1996). This view is not wrong because, in GTM, line-by-line examination of the data is required. The aim of line-by-line coding is to build analysis from data (CHARMAZ, 1983). This way of studying data is an *interactive* process. Besides, you go through categories from codes (from concrete to more abstract) to build

theory. At this point I agree with Kathy CHARMAZ (1983) that categories will emerge through the interaction between the observer and the observed. Categories are grounded in the data thanks to the line-by-line coding but still a category is part of the analytic thinking of the researcher. Raising a code to a category is an analytic process. [6]

Thus, what is important to note is that codes are not always categories. Theory-building occurs by categorization. First-level coding (line-by-line, open and in-vivo coding) is pivotal in the categorization process and it shall go hand in hand with memo writing. Thanks to memo-writing, coding goes beyond indexing. In this context, it is incorrect to draw the conclusion that a systematic approach to analysis excludes an interpretive approach and is contented with indexing. [7]

Applying GTM to my analysis has not prevented me from comparing two CAQDAS packages in terms of interpretative or indexical (coding) approaches to analysis. Kathy CHARMAZ (1983) advises bringing raw data into the memo to preserve the most telling examples. If you are using a CAQDAS package that enables creating links between your data and memo, you will be connected to your data easily. CADQAS packages at the level of theory-building offer both memoing-linking and coding tools in an advanced way. [8]

2.1 My data

The data that is analyzed with the two software packages come from an auto-photographic study conducted in 2007. Eleven university students were asked to answer the question "who are you?" by taking photographs. Auto-photography was first used by Robert ZILLER and his colleagues (ZILLER & LEWIS, 1981; ZILLER & SMITH, 1977) to study self-identity. Later this method was used by DOLLINGER and his colleagues (DOLLINGER & DOLLINGER, 1993, 1997). [9]

Both ZILLER and DOLLINGER used written explanations of photographs and they undertook a quantitative analysis of the data. Carey NOLAND (2006) was the first who combined auto-photography with qualitative interviews. In my study, like NOLAND, I conducted qualitative interviews with each student to talk about the photographs that they had taken. Individual in-depth interviews (lasting between one and two hours) were undertaken to let the students talk about each photograph. [10]

3. First-Level Coding Process with Two CAQDAS Packages

In this section I focus on the coding and memoing tools of the two selected CAQDAS packages. The tools are shown and evaluated under the following three criteria selected according to my methodological approach: closeness to the data, ease of coding and memoing, and the interrelationship among the data, code and memo. To pursue this goal, it is important to introduce the terminology used in these two packages. [11]

3.1 Terminology

Both CAQDAS packages support the coding process in an advanced and similar way. Yet, there are some terminological differences between them: in NVivo 8 the holder of the coded segments is called a "node." Some nodes are called "free nodes" and others "tree nodes." Free nodes cannot have any hierarchical order while tree nodes are nodes that can be arranged hierarchically. In MAXQDA, the holder of the coded segments is called "code." Like NVivo, MAXQDA offers hierarchical organization of codes, the difference is that there is no terminological differentiation regarding the two ways of organizing codes—as a list or as a hierarchy. It is worth to mention here the distinction made by LEWINS and SILVER (2007): *cosmetic* and *functional* hierarchical systems. In a cosmetic hierarchical system, you can prefix the codes to appear together in the main code list. But it does not link the codes. On the other hand, functional hierarchical systems are more than just the cosmetic appearance of the hierarchical structure (p.116). Both NVivo and MAXQDA offer functional coding scheme hierarchy that allows retrieval of all data associated within a hierarchy. [12]

As for the memoing, both offer several tools and these tools have different names. The memoing tools available in these packages are listed in Table 1.

MAXQDA Memoing Tools	NVivo Memoing Tools
MEMO	MEMO
COMMENT	ANNOTATION
TEXT-LINK	SEE ALSO LINK

Table 1: Memoing tools in the two CAQDAS packages [13]

As seen in Table 1, there are other memo-type tools than those termed "memos" in both packages. In qualitative methodology, memos are all kinds of notes taken by the researcher during the whole research process. You can write memos about the whole project, about your data in general, per each data source or part of it, about your analysis, codes etc. [14]

In NVivo, you can create and link memos to your data sources or codes (nodes). Memos cannot function as writing notes for chosen segments of the data source or coded content; instead they are notes for the entire data source or code. When you are working with NVivo and you want to add a specific note for a data segment you can do so by using the "Annotation" tool. [15]

In MAXQDA, you can create and link memos to your entire data source or code or to chosen segments of it. One more tool available in MAXQDA for memoing is the "Comment" tool that allows you to add small notes to individual code assignments. [16]

Briefly, even though the terminology differs, both software packages offer similar functions for memoing. Yet, some critical differences exist between them in terms of how their memoing tools are designed, in their *architecture*. These architectural differences may have an effect on your analytic possibilities (see Section 3.5). [17]

3.2 Interfaces: Coding workspaces

In NVivo, the Detail View Window is used for browsing data sources (documents, pictures, audio or video files). So, the first step is to open the data source and then start reading and coding. During the coding process, you have the category (node) list in the List View window which can be arranged to be at the left side of Detail View. It is also possible to arrange the interface: by hiding areas.

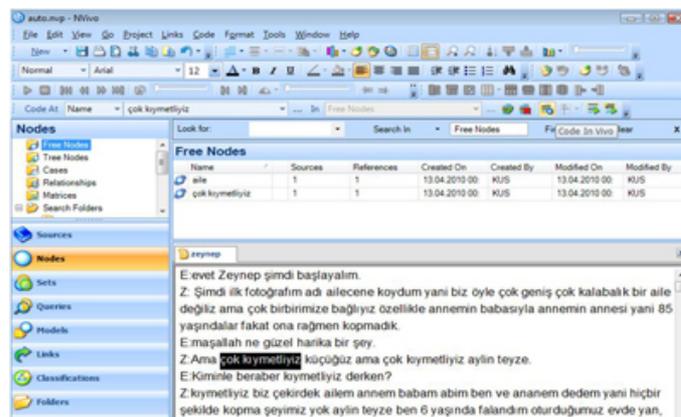


Figure 1: NVivo interface (enlarge this figure [here](#)) [18]

When we look at the MAXQDA interface, we see four windows: Document System, Document Browser, Code System and Retrieved Segments windows. Here also, it is also possible to arrange the interface by hiding some of the windows.

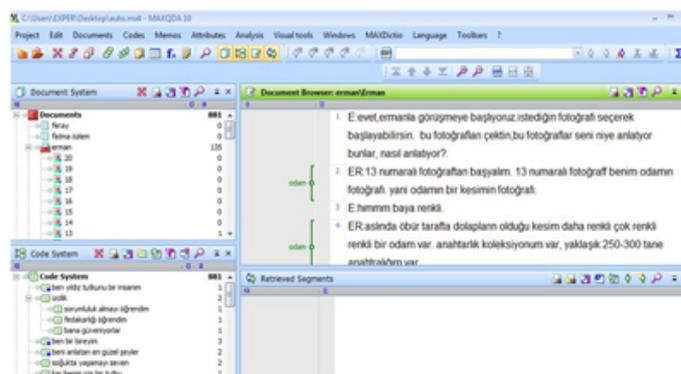


Figure 2: MAXQDA interface (enlarge this figure [here](#)) [19]

3.3 Open coding and in-vivo coding

GTM is based on induction. The data itself constitutes the starting point. You start reading the data line-by-line to understand what is said or done. You make your interpretations based on your data and you start with coding as the first step of categorization. Coding cannot be seen as just indexing or a systematic way of approaching your data. Coding is part of your understanding and interpreting. Proceeding with line-by-line and in-vivo coding prevents us imposing any prior conceptualization. But, the coding process should not be seen as just a way of indexing the data and should be linked to your memo as a container of your reflections. [20]

In an inductive analysis, in-vivo coding is indispensable. "In-vivo code" is a term in GTM used to refer to the codes that you take directly from the participant's discourse. Even the name of the code is derived from the participant's own words. Both NVivo and MAXQDA offer an in-vivo coding function. The basic steps are common in both, and: in general, performing first level coding is very easy to do with either package, both always keeping you close to your data whilst doing so. Nevertheless, the architectural differences—such as how you access the coding tools and how they are designed—are still significant and need to be displayed for further methodological evaluations. [21]

With NVivo

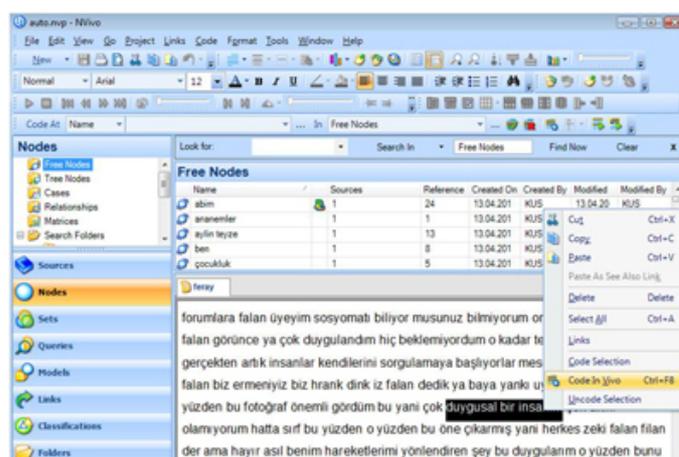


Figure 3: In-vivo coding with NVivo (enlarge this figure [here](#)) [22]

The NVivo category system differentiates between several types of codes, two of which are important in the context of this article: "tree nodes" and "free nodes" (the organization of the category system has been changed in the latest version of the software NVivo 9). All in-vivo codes are by default saved as free nodes. By right clicking on the selected data segment you find the "code in-vivo" option in the contextual menu (Fig.3). [23]

In fact, at any moment you can create a new node using the List View window just by right clicking on it and selecting the option "New node." But when you are

reading the text, you are scrolling down the text with the mouse. When you make a decision to code a part of it, the easier or more comfortable way is to directly select it and right click on it. [24]

List View Window can display one folder at the time (for example Free Nodes or Tree Nodes). This architectural structure for categories in NVivo puts some limits to the way you do your analysis. For example, at any moment you may want to create a hierarchy among your categories. But in NVivo, as long as you have free nodes you cannot do it. First, you must transform these free nodes into tree nodes. This transformation process is quite easy: you just cut and paste on the tree node folder. The problem is that you must open the tree nodes file in List View to create the sub-category. [25]

Thus, from a procedural point of view the architectural design of the category system in NVivo, facilitates a process in which first you code in-vivo which is free from any hierarchical order; and then you organize into categories using Tree Nodes. [26]

With MAXQDA

In MAXQDA when you decide to code a segment of the text you select it and right click on it. Through this way the contextual menu is opened, including the two options for coding: "Code in-vivo" or "Code with a new code" (Fig. 4).

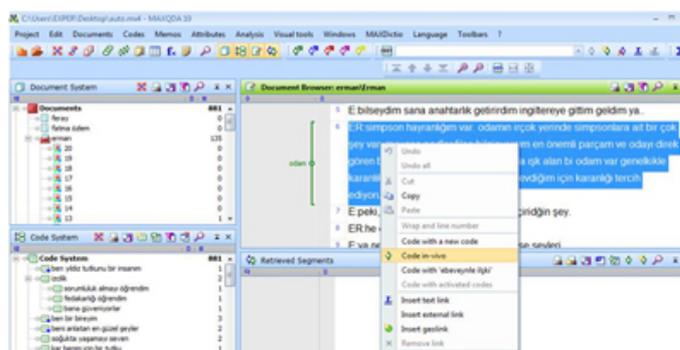


Figure 4: Coding with MAXQDA (enlarge this figure [here](#)) [27]

The Coding System window stores all of the codes regardless of the way you create them. When you select the data segment from the text and you code in-vivo by right clicking on it, your new code appears both on the Code System Window and on the left margin of the selected text. If you want to organize this in-vivo code under a new category, you create a new code in the Code System Window and you drag and drop the in-vivo code under this new code. [28]

3.4 Coding Tool Bar

When you proceed with coding inductively, after a while, you may have a long list of codes and you may want to code a data segment to an existing code from the list. Therefore, it is important to have a look at how the two CAQDAS packages differ in this respect. [29]

With NVivo

To code a data segment to an existing node in NVivo, the easiest way is to select the related data segment and then drag and drop it on the related node in the list. [30]

But it is difficult to catch the right node when the node list is long. Another option is to use the Coding Tool Bar (Fig. 5). After you select the related data segment, you can select the node from the Coding Tool Bar and click on the "Code" icon.



Figure 5: Coding Tool Bar in NVivo [31]

The list of nodes available through the Coding Tool Bar drop-down menu is limited to the nodes that have been recently used. Besides, when moving between tree and free node folders, the drop-down menu for the nodes at the Coding Toolbar becomes inactive until you perform coding on one of the nodes in the tree node list. For this reason, most of the time, it is not functional to use the Coding Tool Bar. [32]

One more option is to use the "Code" menu from the main tool bar. After you select the related data segment, you click on the "Code" menu and go through the contextual menu and click on the option "Code Selection / At Existing Nodes ...". When you click on this option, a Dialog Box will open with the list of all created nodes (Fig. 6). On this Dialog Box, you can select the node or nodes you want to code. This Dialog Box for coding is quite useful especially when you want to quickly apply several codes to one segment.

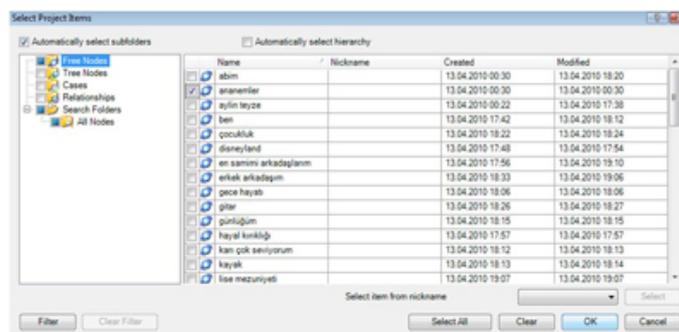


Figure 6: Dialog Box reached from the "Code" menu on the main tool bar in NVivo (enlarge this figure [here](#)) [33]

Using the "Code" menu in the main tool bar is not practical, especially when you are coding line-by-line. I must note here that my evaluation for this point is based on how I use my computer in general. I use the mouse more than the keyboard and for this reason, I don't use shortcuts. Shortly, the way I perform coding is affected by my mouse habit. That is why I always feel comfortable with contextual menus that are available through right-clicking since I am reading the text by scrolling with the mouse. Whenever I want to perform coding I just select the relevant segment from the text and I right-click the mouse. Even though the evaluation of the software architecture in terms of habits related with computer use is beyond the subject of this article, I wanted to add this point as a reflection on the relationship between our habits and architectural design. [34]

At this point, to be fair with NVivo I should add that "Code" menu options can be reached by shortcut keys, which makes coding easy and practical. The Dialog Box is opened directly so that you can apply several codes to the related segment. The Dialog Box option for coding is one of the pros of NVivo's architecture. [35]

With MAXQDA

When you have a long list of codes created inductively and you want to code an existing code from the list of codes, using the Coding Tool Bar in MAXQDA is the easiest way. The dragging and dropping option is available also in MAXQDA but I prefer to use Coding Tool Bar because it is very practical. [36]

When the code list is long, you can use the "Context Search" tool, available on the Code System Window's tool bar, to find the code that you want to assign. [37]

By just clicking on the related code, you will see it in the Coding Tool Bar. Now, you can select the related data segment and you can click on the "code" icon available on the Coding Tool Bar (Fig. 7). From the architectural point of view, the interaction between the Code System (CS) Window and the Coding Tool Bar is quite effective: whenever you click on a code on the CS Window, that code appears on the Code Bar.

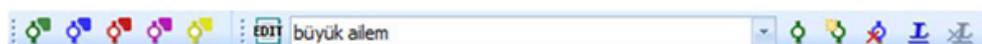


Figure 7: Coding Tool Bar in MAXQDA [38]

The main difference from NVivo is that the MAXQDA Coding Tool Bar functions better. The menu on the tool bar shows any code that you want to see and by just clicking on a code in the Code System Window, the code will appear on the drop-down menu. There is no differentiation between codes like "free / tree node" and all the codes are displayed in one Window. [39]

Regarding the deductive or inductive way of coding, it is correct to say that CAQDAS packages do not put any limit; this is the decision of the researcher. And even the researcher may use both ways at the same time. This is valid also for NVivo 8, *but not very practical*. When you're doing open coding or in-vivo coding, by-

default these nodes are created as free-nodes in NVivo 8. You can then transfer them to the tree-node folder, where you can start to create sub-codes (child nodes). What makes it impractical is the architectural design for category system. Tree-nodes and free-nodes were not shown in the same Window or side by side. Fortunately, this structure has been changed in the latest version of NVivo. [40]

Apart from the architectural difference regarding the category system, both NVivo and MAXQDA offer coding tools that can be easily accessed. [41]

3.5 Adding reflexive notes: Memos, comments, annotations

As I mentioned earlier, memoing is crucial during the first level coding process. As Barney GLASER stated (2004), if you perform coding without memoing then you are not doing GTM. Your decision to create a code and assign a data segment from the text to that code is an interpretative process. "There is always a perception of a perception as the conceptual level rises. We are all stuck with a "human" view of what is going on and hazy concepts and descriptions about it" (p.45). For this reason, an immediate recording of conceptual ideas grounded in the data is required. [42]

In this context, reflexive tools (e.g. comments, memos or annotations) offered by the CAQDAS packages gain essential importance to the GTM approach. I prefer to call memoing tools in CAQDAS packages "reflexive tools" because this term is more inclusive for the software tools. [43]

3.5.1 Memos in NVivo

In NVivo, you can create a memo for a code or for a data source. In addition you can use the "Description" in the "Node properties" dialog box to write your definitions or thoughts. At any moment, you can right click on an item (e.g. node, source) and select "Memo Link / Link to New Memo" from the contextual menu (Fig. 13). The "New Memo" option is also available when you select a data segment and right click on it (Fig. 8).

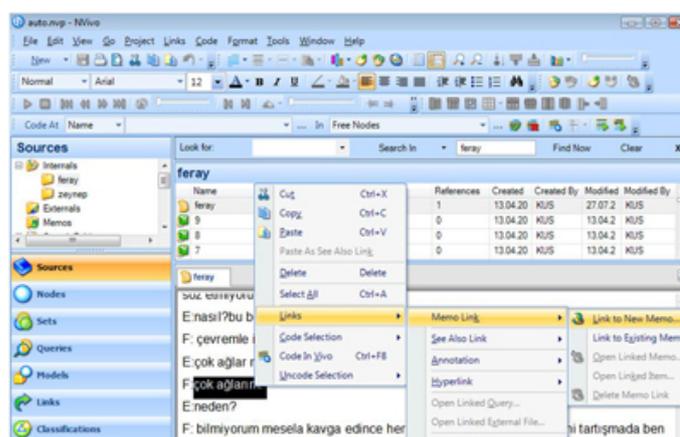


Figure 8: Creating memo with NVivo (enlarge this figure [here](#))

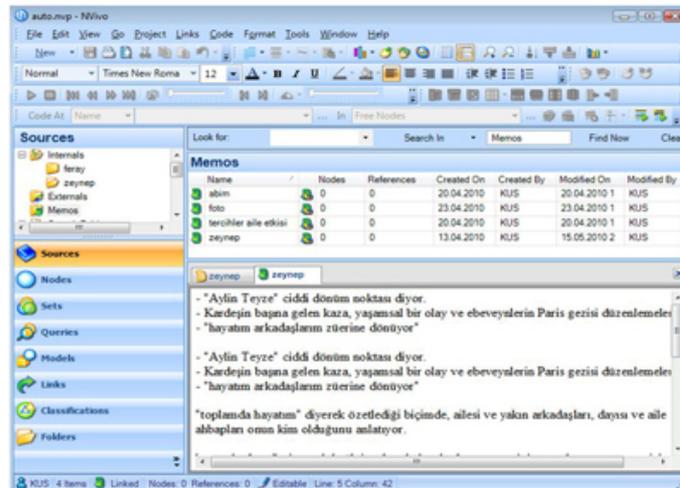


Figure 9: Memo view in NVivo (enlarge this figure [here](#)) [44]

In NVivo, your memo is opened as a new tab on the Detail View Window (Fig. 9). You can jump between your text (data source) and your memo. One negative side is that you do not see your related data and your memo simultaneously without going through the process of undocking both from the main interface. [45]

3.5.2 Annotations in NVivo

As mentioned before, linking a memo to the node or the data source links to the entire item, not to a segment of it—even though you choose to link the memo by selecting a data segment. If you want to attach your memos just for a specific segment of your data, you can create a link to your notes which is called "Annotations" (Fig. 10). The Annotation will appear like a footnote at the end of the data source or node.

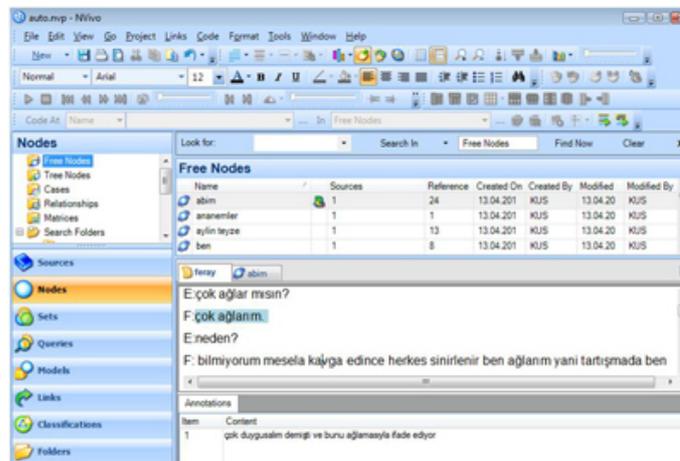


Figure 10: Annotation view in NVivo (enlarge this figure [here](#)) [46]

The data segment, with an annotation attached to it, appears highlighted in blue. In this way, NVivo reminds you that there is an annotation linked to the data

segment. To read what is written in an annotation, you can click on the button "View Annotations" that is available on the tool bar. [47]

3.5.2 "See Also Link" in NVivo

Another reflexive tool in NVivo is the "See Also Link." Thanks to this tool, you are able to insert a link to a segment of your data. This link may be another data source or any item existing in your project or a new item created by you. You use this tool similarly to the other Links tools: you select a data segment and click on "Links" from the main tool bar or just right click on it and select "Links / See Also Link / New See Also Link ..." from the contextual menu. Thus, you are creating a link to an item but not to a segment of it. [48]

In my view, "See Also Link" functions better when you create a link between two different data segments. You can select a data segment, right click on it and select the "Copy" option. Then you can select the other data segment that you want to link, right click on it and select the "Paste as See Also Link" option (Fig. 11). The linked segments will be highlighted in red so that wherever you see the segments (in the original context or in the coded context) they will function as a reminder for you.

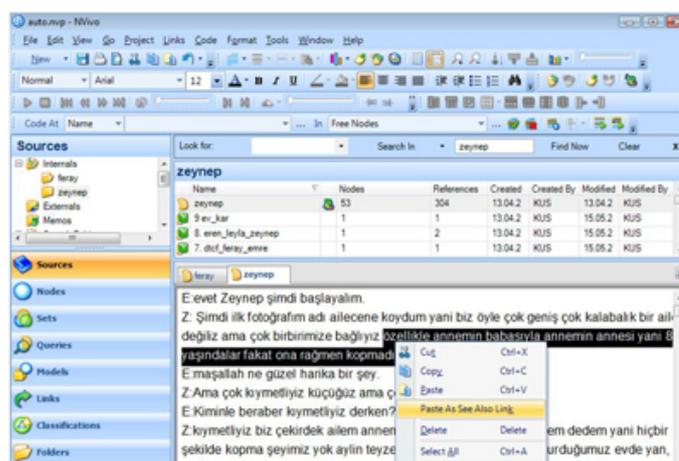


Figure 11: Creating "See Also Link" between data segments (enlarge this figure [here](#)) [49]

3.5.3 Memos in MAXQDA

As I explained before, your memos in MAXQDA can be attached to an entire data source or code but at the same time to a specific data segment. The way you create memo is similar to NVivo: you select a data segment to which you want to create a link, and then you right click on it and select the "New Memo" option from the contextual menu. [50]

When you create a memo, the memo icon appears on the left margin of the selected segment. You have different options to visualize your memo icon, which is quite useful for classification purposes (Fig. 12). The memo is opened as a dialog box and this makes you feel close to the data because, the data source

and the memo dialog box are both in use simultaneously. Besides, you can read what is inside your memo just by putting your cursor on the memo icon (hovering) displayed at the left margin of the selected data segment (Fig. 13). You can also open your memo by double-clicking on it, so that you can also edit it if you need to do so.

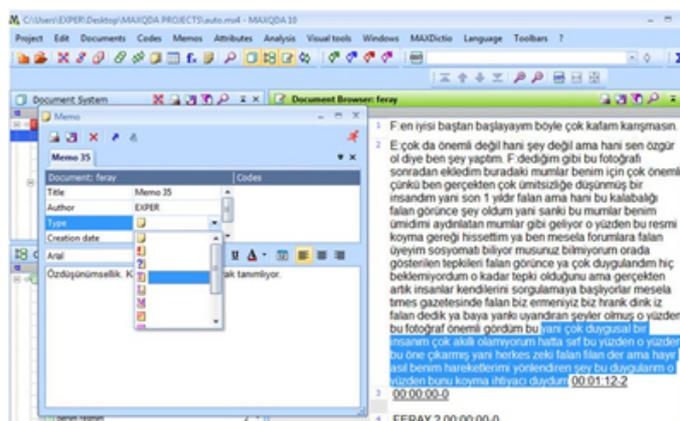


Figure 12: Memo icons in MAXQDA (enlarge this figure [here](#))

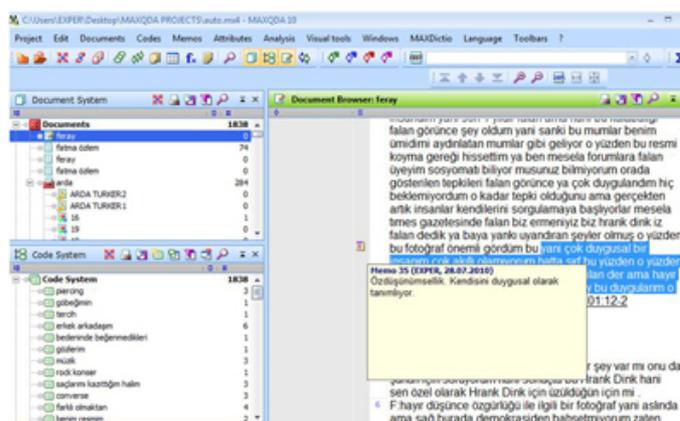


Figure 13: Memo display by hovering in MAXQDA (enlarge this figure [here](#)) [51]

3.5.4 "Text Link" in MAXQDA

Another reflexive tool available in MAXQDA is the "Text Link" tool. It works like the "See Also Link" in NVivo: first, you select a data segment, right-click on it and choose the "Insert text link" option. Second, you select the other data segment that you want to link, you right-click on it and choose "Insert text link" option. The data segments linked in this way are shown in blue color and underlined. [52]

Like in the memo display, MAXQDA allows you to display your Text Link by just putting your cursor (hovering) on the related data segment. [53]

3.5.5 Comments in MAXQDA

MAXQDA offers another reflexive tool called "Comment." This tool is to add short notes to individual code assignments. When you are working on your data source, each time you create a code, the code symbol appears at the left margin of your data source. It displays each code assignment. To add your comment, you double-click on the little circle in the middle of the code symbol, then the dialog box appears. After you write your comment about the code, the little circle in the middle of the code symbol becomes filled. Thus, whenever you see a code symbol with a filled circle, you can just put your cursor on it and display your comment (Fig. 14).

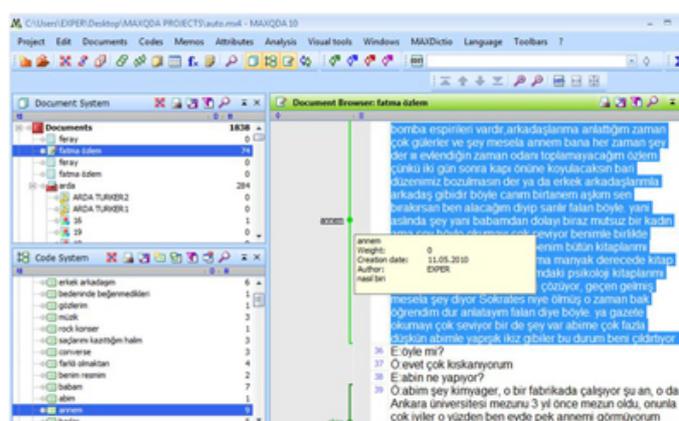


Figure 14: Comment display in MAXQDA (enlarge this figure [here](#)) [54]

I would like to emphasize the architectural difference between the two CADQAS packages. As seen above, all the three reflexive tools in MAXQDA have a display through hovering. Besides, when you are adding your memos or comments, the dialog box is displayed simultaneously with your data source. Based on my third evaluation criterion (*the interrelationship among the data, code and the memo*), the architectural design of MAXQDA is better: your code, memo and comment are displayed at the same time on the left margin of the Document Browser Window. This means that whatever you are doing (coding or adding notes/memoing) on your original data source, it is always shown at the left side of the source. This architectural design supports grounded theory analysis as an interactive process (CHARMAZ, 1983) very well. Being able to add memos to each code assignment strongly supports analytical thinking about each "line," which is crucial in GTM. [55]

In NVivo, displaying nodes at the (right) margin of your text is also possible. This way of displaying is called "Coding Stripes" and for this the "Show Coding Stripes" button is available on the tool bar. To display your notes about a specific data segment, you click on the "View Annotations" button which is also available on the tool bar. Annotations are displayed at the bottom of the data source. MAXQDA's architectural superiority is that you do not need to click on a button to have a display of your codes or memos at the margin. Also worth mentioning is that the hovering option enables easier access to any memo content. You can

also add special notes to the code assignments. All of these notes and codes are simply shown at the left margin of your data source. This design simplicity is a positive point. [56]

4. Reorganizing the Codes: Checking Memos, Annotations, Comments

In GTM, selective coding begins after you discover the core category (GLASER, 2004). Discovering the core variable requires an examination of your memos and codes and searching for reoccurring patterns in the data. Thus, when I am speaking of reorganizing the codes, it includes the first level coding process. [57]

At this stage, I need to have a bird's eye view of all the codes at the same time as my memos, comments and annotations. [58]

With NVivo

In NVivo, the List View of Nodes also displays the memos linked to each node. To open the linked memo, you must right click on the item and select the "Memo Link / Open Linked Memo" option from the contextual menu (Fig. 15). Remember that, linked memos are not for a specific segment inside the node; it is for the entire node. Notes for a specific segment held in Annotations in NVivo cannot be seen when examining the node list.

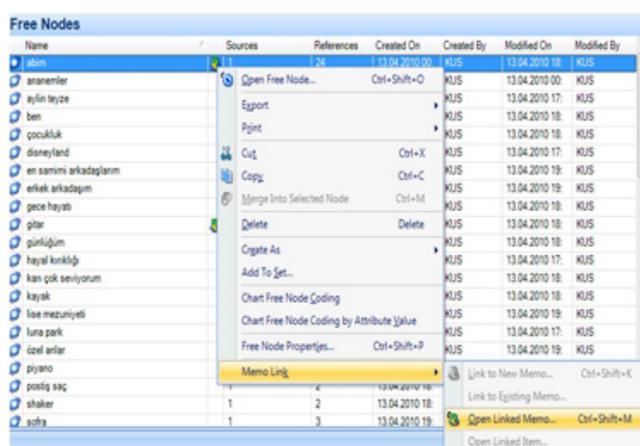


Figure 15: List View of Nodes in NVivo (enlarge this figure [here](#)) [59]

If you want to see coded segments within nodes externally, you must select the Nodes you want from the Node List, then right click on them to Export. This way you can have a HTML (or Word) view of the selected nodes with their contents and even with the linked memos and annotations (Fig. 16). [60]

To examine node content inside your project, you can just double-click on a node. NVivo will display the coded content of the node in the Detail View Window. To display the coded content inside your project is better when you are at the stage of checking, thinking and recoding.



Figure 16: HTML view for a node content in NVivo (enlarge this figure [here](#)) [61]

Node content opened in Detail View Window can be opened separately (Fig. 17), so that you can have two windows (one for the project and one for the node content) displayed simultaneously. When you want to open the linked memo for that node, you can click on the "Links" button on the main tool bar of the node window and then select the "Memo Link / Open Linked Memo ..." option. [62]

The linked memo for the node will be opened in the project window of NVivo. To read (or edit) it you can jump into the project window. To see your annotations and See Also Links inside the node, you can click on the "View" menu on the main tool bar and select to view both. Thus, you will see both the linked annotations and the See Also Links at the bottom of the node window (Fig. 18). By just clicking on them, NVivo will display their location (linked data segment).

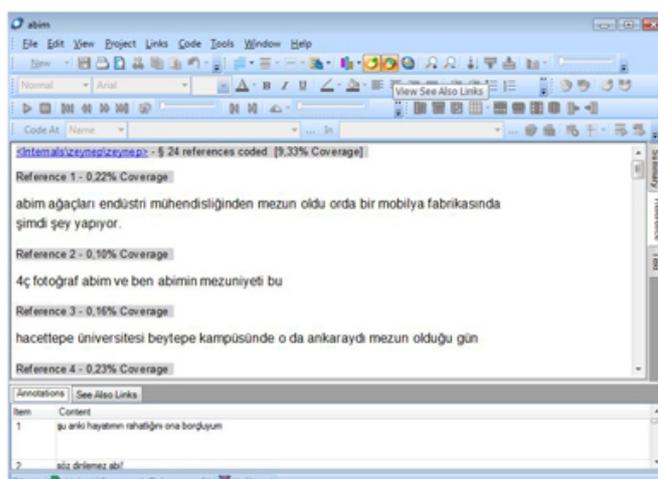


Figure 17: Viewing annotations and See Also Links of a node in NVivo (enlarge this figure [here](#)) [63]

When you want to see all your linked items (memos, annotations and See Also Links), you can open the "Links" file content. All the linked items will be listed in the List View Window (Fig. 18).

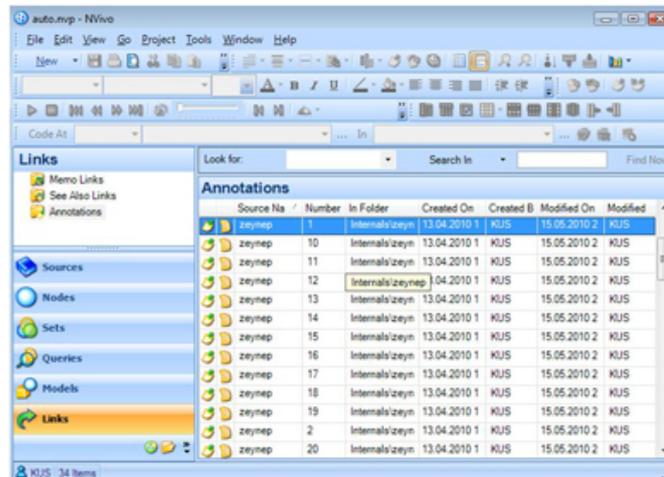


Figure 18: Links (annotations) displayed in Detail View Window in NVivo (enlarge this figure [here](#)) [64]

By just double-clicking on an annotation link, NVivo will open the link in Detail View Window. But if the memo folder is displayed in List View Window, then you must right click on the linked data source and select the "Open linked memo" option from the contextual menu. [65]

With MAXQDA

In MAXQDA, the best way of examining your code list is to click on the "Codes" button on the tool bar and to select "Overview of Coded Segments." The Coded Segments Window will be opened (Fig. 19). Here, you see which code is from which data source, the coded segments inside each code and the Comments. The coded segments of each code have a Preview column and when you click on a code segment, the whole segment is displayed on the top of the window. Memos are not displayed in the Coded Segments Window, but all the Comments are displayed. Moreover, you can add or edit Comments in this Window.

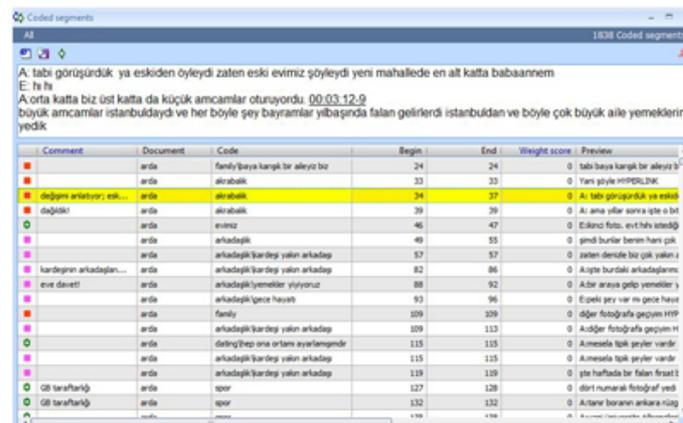


Figure 19: Coded segments window in MAXQDA (enlarge this figure [here](#)) [66]

If you are doing Grounded Theory, the Coded Segments Window is a unique tool to help you find the core category. This is because you have a table view of all coded segments with the Comment for the code. More clearly, *your data, your code and your Comment are available to you simultaneously*, which makes it very practical to reflect on (reread, recode, etc.) the coded segments. When you click on one segment to display what is inside, MAXQDA displays this segment in its original context automatically in the Project Window. If you have a big screen you can display the two MAXQDA Windows simultaneously (Fig. 20). This means, you also have your Memo (if that segment has one) at hand. You can reread or edit, your memos, comments, data and your codes in a very interactive and very user-friendly way.

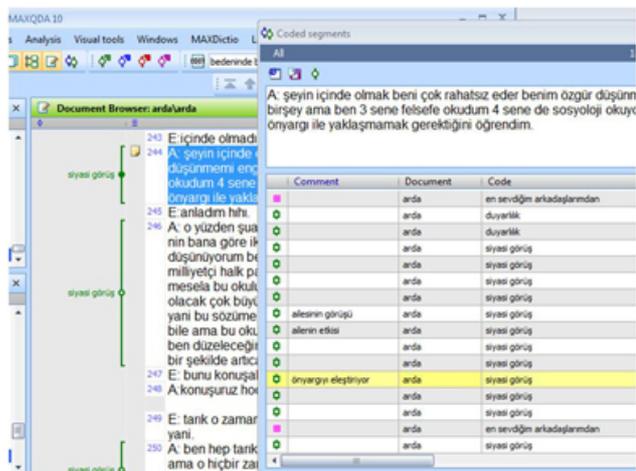


Figure 20: Project Window and the Coded Segments Window in MAXQDA (enlarge this figure [here](#)) [67]

When you want to see the list of all your memos, you can click on the "Memos" button on the tool bar and select the "Memo Manager" option. Again, a new window will open to display all your memos with their symbols, preview, linked data segment, etc. (Fig. 21). Here you can click on a memo to display what is written inside. MAXQDA will display your memo in its original location in the project window.

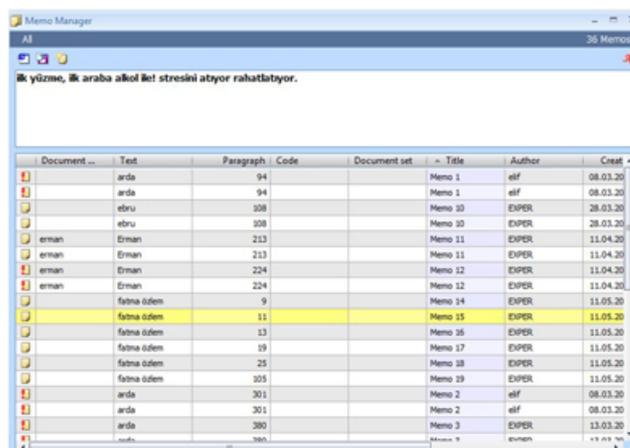


Figure 21: Memo Manager Window in MAXQDA (enlarge this figure [here](#)) [68]

5. Conclusion

My focus in this article was to compare two CAQDAS packages based on a specific methodological approach: Grounded Theory. Even though GTM is considered as supporting a *systematic approach* to qualitative data analysis, I have highlighted that GTM cannot be reduced simply to coding. Constant recording of the ideas, interpretations of the researcher—i.e., memoing—must be performed at all stages of the whole research and analysis. As GLASER stated, memoing is never done: "Early on memos arise from constant comparison of indicators to indicators, then indicators to concepts. Later on memos generate new memos, reading literature generates memos, sorting and writing also generate memos—memoing is never done" (2004, p.64). [69]

Thus, I focused on the first-level coding process in GTM in order to compare the two CAQDAS packages in terms of open coding and memoing. Both NVivo and MAXQDA are theory-building software packages and offer many similar tools for users. The main difference between them in relation to the comparison presented in this article is the architectural design of accessing and working with codes and memos. Based on my experiences as a user and trainer, I am aware of the implicit effect of software architecture on the way researchers conduct analysis. This effect can only be made visible by focusing on the software architecture from a specific methodological approach. [70]

Stating the importance of memoing as well as coding, I compared NVivo and MAXQDA open coding and memoing tools. Both of them offer similar tools for reflexive thinking or memoing. The main difference between them is the architecture of these reflexive tools (Fig. 22). In this article, I haven't given any specific example to show the contribution of MAXQDA on my GTM analysis. But I would like to emphasize that in general, MAXQDA motivated me to interact with my data as much as I can through its easy and interactive design.

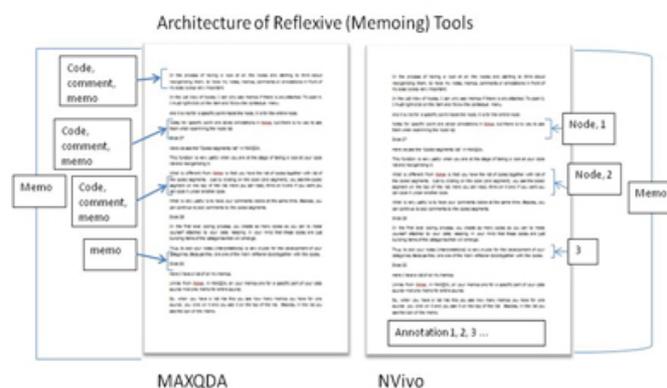


Figure 22: Architecture of the reflexive tools in NVivo and MAXQDA (enlarge this figure [here](#)) [71]

When comparing the coding and the memoing tools of the two CAQDAS packages, my criteria for the evaluation of these tools were: closeness to the

data, ease of coding and memoing, and *the interrelationship among the data, code and the memo* which were derived from my methodological approach. Thus, all my interpretations or judgments on the two software packages are bounded with the chosen methodological approach. [72]

In my comparison, I came to the conclusion that MAXQDA supports the interrelationship among the data, code and memo better than NVivo. As a user applying GTM approach, I found the memoing tools in MAXQDA better fitting to my needs for theory building purposes. The ease of reaching and using memoing tools enhanced my interaction with the data. In MAXQDA architecture, codes and memos are shown together at the left margin of the data. In addition, the Comment tool allows adding memos to each code assignment and different icons are offered as a way to classify memos if needed. Furthermore, all the activities such as creating or editing a code or memo can be done in the same Window. Thanks to this simple design, I was able to move between many memos as well as codes and this is why my interaction with the data increased. In MAXQDA, memoing is as easy as coding and the memoing tools function is based on the same logic as coding because each coded segment can have a specific memo attached to it. In sum, this architecture encourages you to use many kinds of memo and by doing so not only does it increase the interaction between the data and the researcher but also it supports self-reflexivity which are the critical points when performing a grounded theory analysis. [73]

But one thing is for sure that, in general, the contribution of the CAQDAS packages to Grounded Theory cannot be denied. I believe that grounded theory analysis can be performed better with the help of software because you are always working close to your original data. Even if you are at a high level of categorization, with one click you can jump into your original data source to see the coded segment in its original context. The software also makes it much easier to perform line-by-line coding. Finally, tools for linking enhance the interactivity that is needed between the analyst and the data. [74]

The decision of choosing qualitative software is not an easy one. I think that methodological evaluations of the software architecture will be more helpful to users than classifying the software packages according to a list of tools offered by them. For this reason, more user-based evaluations are needed. [75]

References

Charmaz, Kathy (1983). The logic of grounded theory. In Robert Emerson (Ed.), *Contemporary Field Research: A collection of readings* (pp.109-126). Boston, MA: Little, Brown.

Charmaz, Kathy (2006). *Constructing grounded theory*. London: Sage.

Coffey, Amanda; Holbrook, Beverley & Atkinson, Paul (1996). Qualitative data analysis: Technologies and representations. *Sociological Research Online*, 1(1), <http://www.socresonline.org.uk/1/1/4.html> [Accessed: September 25, 2010].

Dollinger, Stephen J. & Dollinger, Stephanie C.M. (1993). Identity, self and personality: II. Glimpses through the autophotographic eye. *Journal of Personality and Social Psychology*, 64, 1064-1071.

Dollinger, Stephen J. & Dollinger, Stephanie C.M. (1997). Individuality and identity exploration: An autophotographic study. *Journal of Research in Personality*, 31, 337-354.

[Fielding, Nigel](#) & Lee, Raymond (2007). Honoring the past scoping the future. Paper presented at CAQDAS 07: *Advances in Qualitative Computing Conference*, Royal Holloway, University of London, 18-20 April.

[Glaser, Barney G.](#) with the assistance of Judith Holton (2004). Remodeling grounded theory. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 5(2), Art. 4, <http://nbn-resolving.de/urn:nbn:de:0114-fqs040245> [Accessed: August 12, 2008].

Kelle, Udo (1997). Theory building in qualitative research and computer programs for the management of textual data. *Sociological Research Online*, 2(2), <http://www.socresonline.org.uk/2/2/1.html> [Accessed: July 1, 2009].

Lewins, Ann & Silver, Christina (2007). *Using software in qualitative research: A step-by-step guide*. London: Sage.

Noland, Carey M. (2006). Auto-photography as research practice: Identity and self-esteem research. *Journal of Research Practice*, 2(1), Article M1, <http://jrp.icaap.org/index.php/jrp/article/view/19/50> [Accessed: January 15, 2007].

Ziller, Robert C. & Lewis, Douglas (1981). Orientations: Self, social and environmental percepts through auto-photography. *Personality and Social Psychology Bulletin*, 7, 338-343.

Ziller, Robert C. & Smith, Dale E. (1977). A phenomenological utilization of photographs. *Journal of Phenomenological Psychology*, 7, 172-182.

Author

Elif KUŞ SAILLARD is a sociologist at Ankara University. She has been using qualitative data analysis software since 2002. She published two books about the use of software in qualitative data analysis. She is teaching NVivo and MAXQDA in workshops.

Contact:

Elif Kuş Saillard

Sociology Department
Ankara University, Faculty of Letters
Sihhiye 06100 Ankara
Turkey

Tel.: +90 312 310 32 80 / 1721

Fax: +90 312 310 57 13

E-mail: kus@humanity.ankara.edu.tr

Citation

Kuş Saillard, Elif (2011). Systematic Versus Interpretive Analysis with Two CAQDAS Packages: NVivo and MAXQDA [75 paragraphs]. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 12(1), Art. 34, <http://nbn-resolving.de/urn:nbn:de:0114-fqs1101345>.