Appendix E: Qualitative Data Analysis

This document was originally part of the manual for The Ethnograph v4. It also exists in the manual for The Ethnograph v5 as Appendix E.. You can freely download, copy, print and disseminate this document if 1) the copyright notice is included, 2) you include the entire document, and 3) you do not alter the document in any way. This permission does not extend to the Manual itself. Only to this electronic version of Appendix E. The information in this document represents some, but not all, of the ideas of the developer of The Ethnograph. I reserve the right to revise and change my ideas as I continue to develop them.

Introduction

This appendix is an essay on the basic processes in qualitative data analysis (QDA). It serves two purposes. First it offers some insights into the ideas and practices from which The Ethnograph emerged and continues to evolve. Second, it is also a simple introduction for the newcomer of QDA.

A Process of Noticing, Collecting and Thinking

The first part of this appendix describes QDA as a process of *Noticing*, *Collecting*, and *Thinking* about interesting things. The purpose of this model (**Figure 1**, page E2) is to show that there is a simple foundation to the complex and rigorous practice of QDA. Once you grasp this foundation you can move in many different directions.

The idea for this model came from a conversation with one of my former teachers, Professor Ray Cuzzort. Ray was teaching an undergraduate statistics course and wanted to boil down the complexity of statistics to a simple model. His solution was to tell the students that statistics was a symphony based on two notes: means and standard deviations. I liked the simplicity and elegance of his formulation and decided to try and come up with a similar idea for describing QDA. The result was the idea that QDA is a symphony based on three notes: Noticing, Collecting, and Thinking about interesting things.
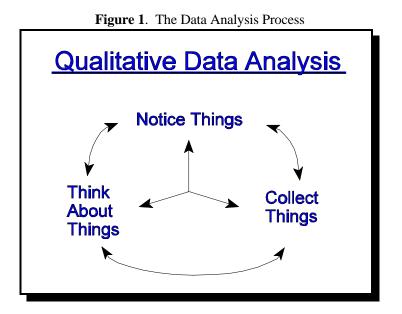
While there is great diversity in the practice of QDA I would argue that all forms of QDA are based on these three "notes."   In the first section of this Appendix I explain this model, introduce the jigsaw puzzle analogy, and offer examples of how the basic QDA model is represented in the writings of QDA methodologists and researchers, and then present alternatives to the jigsaw puzzle analogy.

A General Model of QDA

The second part of this Appendix presents a more complex model of the process of QDA (**Figure 3**, page E13). This model incorporates the simpler model. It shows how the basic procedures of The Ethnograph mesh with the basic model, and how the analytic process unfolds and develops over time.

QDA: A Model of the Process

Analyzing qualitative data is essentially a simple process. It consists of three parts: Noticing, Collecting, and Thinking about interesting things. **Figure 1** represents the process and the relationships among its parts.

Figure 1. The Data Analysis Process

Qualitative Data Analysis

Notice Things

Think
About
Things

Collect
Things

As **Figure 1** suggests, the QDA process is not linear. When you do QDA you do not simply Notice, Collect, and then Think about things, and then write a report.   Rather, the process has the following characteristics:

- **Iterative and Progressive:** The process is iterative and progressive because it is a cycle that keeps repeating. For example, when you are *thinking* about things you also start *noticing new things* in the data. You then *collect* and *think* about these new things. In principle the process is an infinite spiral.

- **Recursive:** The process is recursive because one part can call you back to a previous part. For example, while you are busy *collecting* things you might simultaneously start *noticing* new things to *collect*.

- **Holographic:** The process is holographic in that each step in the process contains the entire process. For example, when you first *notice* things you are already mentally *collecting* and *thinking* about those things.

Thus, while there is a simple foundation to QDA, the process of doing qualitative data analysis is complex. The key is to root yourself in this foundation and the rest will flow from this foundation.

Noticing, Collecting, Thinking about Things

In the next sections I further elaborate on the *notice, collect, think* process. The primary vehicle is the analogy of solving a jigsaw puzzle.   Then I show some of the ways in which the notice, collect, think process has been expressed in the writings of qualitative social scientists. Finally, I explore two alternative analogies. One is the "multi threaded DNA" analogy (Agar, 1991). The other is a map analogy based on the ideas of "topographical" maps and "ad hoc" maps.

1. Noticing Things (and Coding Them)

There are many different perspectives on: 1) the kinds of things that you can, and should, notice in your data, and 2) how you should go about the process of noticing those things. But behind these differences there is the common and simple practice of going out into the world and noticing interesting things.

Two Levels of Noticing

On a general level, *noticing* means making observations, writing field notes, tape recording interviews, gathering documents, etc. When you do this you are *producing a record of the things that you have noticed.*

Once you have produced a record, you focus your attention on that record, and notice interesting things in the record. You do this by *reading* the record. In fact, you will read your record many times. *As you notice things in the record you name, or "code," them.* You could simply call them A, B, C, etc., but most likely you will develop a more descriptive naming scheme.

Coding Things

Coding data is a simple process that everyone already knows how to do. For example, when you read a book, underline or highlight passages, and make margin notes you are "coding" that book. Coding in QDA is essentially the same thing. For now, this analogy is a good place to start.

As you become more experienced in QDA you learn that QDA "coding" is also more than this. Further, you will learn the difference between codes as heuristic tools and codes as objective, transparent representations of facts (Kelle and Seidel, 1995). In this essay I treat codes as heuristic tools, or tools to facilitate discovery and further investigation of the data. At the end of this chapter I address the objectivist-heuristic code continuum.

2. Collecting and Sorting Instances of Things

Pieces of a Puzzle

As you notice and name things the next step is to collect and sort them. This process is analogous to working on a jigsaw puzzle where you start by sorting the pieces of the puzzle. For example, assume you have a puzzle picture with a tree, a house, and sky.   A common strategy for solving the puzzle is to identify and sort puzzle pieces into groups (e.g., frame pieces, tree pieces, house pieces, and sky pieces). Some of the puzzle pieces will easily fit into these categories. Others will be more difficult to categorize. In any case, this sorting makes it easier to solve the puzzle.   *When you identify piece, you are noticing and "coding" them. When you sort the pieces you are "collecting" them.*

Of course this analogy differs in important ways from the QDA analysis process. For example, in QDA you don't always have a final picture of the puzzle's solution. Also, in QDA the puzzle pieces are usually not precut. You create the puzzle pieces as you analyze the phenomena. None the less, the jigsaw puzzle analogy captures some important attributes of the QDA process.

A useful definition of the QDA process, and one that seems to fit well with the jigsaw puzzle analogy, comes from Jorgensen (1989).

> Analysis is a breaking up, separating, or disassembling of research materials into pieces, parts, elements, or units. With facts broken down into manageable pieces,

> the researcher sorts and sifts them, searching for types, classes, sequences, processes, patterns or wholes. The aim of this process is to assemble or reconstruct the data in a meaningful or comprehensible fashion   (Jorgensen, 1989: 107).

A similar idea is expressed by Charmaz (1983). For Charmaz, who works in the "grounded theory" tradition, the disassembling and reassembling occurs through the "coding" process.

> Codes serve to summarize, synthesize, and sort many observations made of the data....coding becomes the fundamental means of developing the analysis....Researchers use codes to pull together and categorize a series of otherwise discrete events, statements, and observations which they identify in the data (Charmaz, 1983: 112).

> At first the data may appear to be a mass of confusing, unrelated, accounts. But by studying and coding (often I code the same materials several times just after collecting them), the researcher begins to create order (Charmaz, 1983: 114).

A concrete example of this processes occurs in Freidson's (1975) *Doctoring Together*. This passage shows how the process moves back and forth between the *noticing* and *collecting* parts of the process. I have "coded" this example to highlight this movement.

Noticing: ...we had carried out some 200 separate interviews...and had them transcribed....Each interview was *read*, and sections of them which seemed to be distinct incidents, anecdotes, or stated opinions about discrete topics....were then typed on 5 x 7 McBee-Keysort cards on which were printed general topical categories to guide coding.

Collecting: Buford Rhea then *read* all the cards and tentatively classified them into the simple content categories we had decided upon in advance.

Noticing: He then *read* them again so as to test, revise, and refine the initial gross classification....

Collecting: . .all cards bearing on some general substantive topic such as "patient relations" were removed from the total set of cards and put together in a pack.

Noticing: All the cards in that large pack of between 800 and 1,200 were then read one by one....

Collecting: ...as they were read, the cards were sorted into preliminary topical piles. (Freidson, 1975: 270-271).

Analysis is More than Coding, Sorting and Sifting

The previous section suggests that disassembling, coding, and then sorting and sifting through your data, is the primary path to analysis. But as Michael Agar (1991) rightly cautions, intensive data coding, disassembly, sorting, and sifting, is neither the only way to analyze your data, nor is it necessarily the most appropriate strategy. I agree with this point. Later I will discuss Agar's alternatives and suggest that they also fit the *notice, collect, and think* process.

3.   Thinking about Things

In the thinking process you examine the things that you have collected.   Your goals are: 1) to make some type of sense out of each collection, 2) look for patterns and relationships both within a collection, and also across collections, and 3) to make general *discoveries* about the phenomena you are researching.

Examining the Pieces of a Puzzle

Returning to the jigsaw puzzle analogy, after you sort the puzzle pieces into groups you inspect individual pieces to determine how they fit together and form smaller parts of the picture (e.g., the tree part or the house part). This is a labor intensive process that usually involves a lot of trial and error and frustration.

A similar process takes place in the analysis of qualitative data. You compare and contrast each of the things you have noticed in order to discover similarities and differences, build typologies, or find sequences and patterns. In the process you might also stumble across both *wholes* and *holes* in the data.

Problems with the Jigsaw Puzzle Analogy

While the jigsaw puzzle approach to analyzing data can be productive and fruitful, it also entails some risks and problems. Experienced qualitative social scientists have always been aware of the potential problems, and organize their work to minimize the adverse effects. For example, Wiseman, who does code data, points out that the simple act of breaking down data into its constituent parts can distort and mislead the analyst.

> ...a serious problem is sometimes created by the very fact of organizing the material through coding or breaking it up into segments in that this destroys the totality of philosophy as expressed by the interviewee--which is closely related to the major goal of the study (Wiseman, 1979: 278).

Part of the solution to this problem is as follows:

> *To circumvent this problem*, taped interviews were typed in duplicate. One copy was cut apart and affixed, by subject matter, to hand sort cards and then further cross-coded by coders....*A second copy of the interview was left intact to be read in its entirety* (Wiseman, 1979:278, my emphasis).

In short, Wiseman protects her analysis by working back and forth between the parts and the whole of her data.

Alternatives to the Jigsaw Puzzle Analogy

One general problem with the jigsaw puzzle analogy is that it assumes that the best way to proceed is by intensive and inclusive coding of the data. It assumes that analytic discoveries directly follow from the process of coding and then sorting and sifting coded data. As I have already noted, and will discuss later, while this can be a good way to proceed it is not always the most appropriate or useful approach to analyzing qualitative data.

Examples of Noticing, Collecting, Thinking

The general process of *Noticing, Collecting*, and *Thinking* about things is reflected in many works which describe and discuss the practice of analyzing qualitative data. Four examples are presented below. In each example I have "coded" the text by breaking it up and inserting the terms Noticing, Collecting and Thinking into the text. This is one way of creating a "topographic" map of the text. While the fits are not always perfect, each statement is consistent with the model.

Example 1

The first example comes from a description of QDA by Danny Jorgenson (1989). While this example repeats a previously quoted passage, this time I specifically identify the parts of the quote that correspond to the parts of the QDA process.

Noticing: Analysis is a **breaking up, separating, or disassembling of research materials into pieces, parts, elements, or units**.

Collecting: With facts broken down into manageable pieces, the researcher **sorts and sifts them**,

Thinking: **searching for types, classes, sequences, processes, patterns, or wholes**. The aim of this process is to assemble or reconstruct the data in meaningful or comprehensible fashion (Jorgenson, 1989: 107, my emphasis).

Example 2

Another example comes from a discussion of grounded theory by Corbin and Strauss (1990).

Noticing/

Collecting: Open Coding is the part of analysis that pertains specifically to **the naming and categorizing** of phenomena through close examination of the data. ...During open coding the **data are broken down into discrete parts,**

Thinking: **closely examined, compared for similarities and differences**, and **questions are asked** about the phenomena as reflected in the data (Corbin and Strauss, 1990: 62, my emphasis).

Example 3

A more concrete description of the process is provided by Schneider and Conrad (1983). They describe the analysis of interviews they had collected in an interview study of epilepsy. In this example the codes emerged from the data.

Noticing: We began coding the interviews by **reading carefully** a sample of the transcripts **to develop substantive and general topic codes**....We then photocopied the original transcripts, **marked each appropriate line or section with the code in the margin**,

Collecting: and **cut up and filed the pieces of paper according to the codes**....

Thinking: Fairly early in our project **it became apparent** that the medical perspective on epilepsy did very little to describe our respondents' experience (Schneider and Conrad, 1983:242, my emphasis).

Example 4

Finally, Spradley (1979) sketches the traditional process of anthropological field work. In this example, the noticing process is presented both on the general level of gathering data, and on the particular level of examining the data. "Sorting through field notes" implies noticing something

that can then be collected.

Noticing:	And so the ethnographer started hanging around, **watching, listening, and writing things down**...In a few months, the stack of field notes about what people said and did grew quite large....
Noticing/ Collecting:	The field work period drew to a close and the ethnographer returned home with notebooks filled with observations and interpretations. **Sorting through field notes** in the months that followed....
Thinking:	the ethnographer **compared, contrasted, analyzed, synthesized, and wrote** (Spradley, 1979: 227, my emphasis).

Alternatives to the Jigsaw Puzzle Analogy

The jigsaw puzzle analogy assumes that analysis simply emerges out of coded, sorted and sifted data. But this is not always the case. Many times your *analytical discoveries are only facilitated by, rather than transparently derived from*, the way in which you have coded your data. The risk in following the jigsaw puzzle analogy too closely is that you might get deeply into the pieces and end up *finding the codes but losing the phenomena*.

For example, if you just have the names of streets in a city, you know something about the city. For example, in some cities many streets are named after presidents. In others many streets are named after trees. But simply knowing the names of the streets doesn't necessarily tell you much about the layout of the city, or how to get around in the city. For this you need a concrete representation of the streets in relationship to each other. Further, you need to be able to distinguish between neighborhood streets versus main traffic streets. Similarly, just having a collection of code words, or collections of coded segments of data, does not tell you everything you want and need to know about your data, and how the pieces of your data fit together.

"A little bit of data and a lot of right brain"

Michael Agar (1991) argues that while coding, segmenting, sorting and sifting data can be productive and useful strategies, there are other equally important strategies. The first alternative is to start by intensively examining a small bit of data, rather than intensively coding data.

> My point at the moment is just that this critical micro-level work requires looking at a few detailed passages, over and over again, doing the dialectic dance between an idea about how text is organized and a couple of examples, figuring out what I was looking at, how to look at it, and why (Agar, 1991: 190).

> That critical way of seeing, in my experience at least, comes out of numerous cycles through a little bit of data, massive amounts of thinking about that data, and slippery things like intuition and serendipity (Agar, 1991: 193).

> For that, you need *a little bit of data, and a lot of right brain* (Agar, 1991: 194).

The question is, how do you come up with that "little bit of data?" Obviously you start by reading and rereading the data record. In the process you *notice* a few interesting things. You then *collect* one or more of these things and intensively *think* about them. So you are still within the basic model. I would like to carry this a step further and claim that when you identify and extract the

segment with which you want to work, you are in fact coding the data. The difference is that you are not intensively coding, nor are you consumed by the sorting and sifting process.

Parts in Context, "Patterns Among the Patterns"

Agar's second alternative is to look at coded, but unsorted, passages of data. This alternative seems to be consistent with intensively coding data. But it bypasses the sorting and sifting process. It goes directly from coding to discovery. The analysis is not built on sorting and sifting. Agar describes the process in the following way.

> I need to lay out a couple of stretches of transcript on a table so I can look at it all at once. Then I need to mark different parts in different ways to find the pattern that holds the text together and ties it to whatever external frame I'm developing. The software problem here would be simple to solve. You'd need to be able to quickly insert different colored marks of different kinds at different points so you could see the multiple connections across the text all at once, sort of a multi-threaded DNA laid on the text so you could look at the patterns that each thread revealed and then the patterns among the patterns (Agar, 1991:193).

Here Agar is describing a process where you read and *notice* many things in the data record. Then you focus your attention on one part of the "coded" data record. This part can be chosen at random or deliberately.

Intensively Analyzing a Small Piece of the Data

Agar argues, in general, that QDA computer programs, such as The Ethnograph, are primarily oriented toward segmenting and sorting data, breaking down wholes into parts, and focusing attention on the collections of parts, at the expense of the wholes from which they come. Consequently QDA software biases the analyst toward segmenting and sorting, and away from intensive analysis of small bits of data, and away from viewing the parts in context.

I would argue that The Ethnograph, at least, does in fact facilitate the two analytic alternatives proposed by Agar. In fact, The Ethnograph is unique in its ability to approximate Agar's "multi-threaded DNA" model.
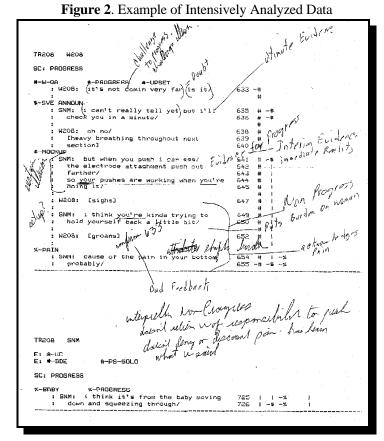
In regard to the "little bit of data, lot of right brain" strategy, the coding and collecting of segments of data can provide the foundation for the process of intensive analysis of a small bit of data. For example, in order to find a piece of data to intensively analyze, Agar is still going through the process of *noticing* and *collecting* a piece of the data. When using QDA software the preliminary coding, and preliminary sorting and sifting, can generate pieces that become candidates for the intensive analysis described by Agar. The trick is to *avoid intensive coding early in the analytic process.*

But even if you have done intensive coding you can always change the analytical direction, and shift your attention to a single piece of data for intensive analysis. In short, one approach does not preclude the other. In fact they can complement each other, and software can facilitate the shift to and from intensive analysis.

An example comes from my own experience. A group of colleagues and I were analyzing data collected during a study of interactions between nurses and women during the process of giving birth. One interesting thing we noticed in our data was that the labor room nurse periodically talked about making "progress" during the birth process. We collected instances of "progress" talk and scheduled an analysis meeting on the topic.

During an analysis meeting a team member would present one or two data segments. We also had access to the original coded transcript and the video from which it was transcribed. We would spend several hours analyzing and thinking about those segments. Each team member had a printout of the segment and would cover it with notes, thoughts, and scribbles. An example printout is shown in **Figure 2**. At the end of the session we would write up a preliminary memo summarizing our work.

During analysis our attention was not restricted to the particular segment. For example, we might also examine and compare other "progress" segments with the segment we were analyzing. We would also look at the transcript from which the segment came so that we could place our analysis in the larger context from which the segment came. This context was not simply the immediately adjacent text within which the segment was embedded, but the entire event of which it was a part.

Figure 2. Example of Intensively Analyzed Data

This type of analytic process was not focused on gross analysis and summarization of a category of the data. Rather, it emerged out of preliminary coding and followed Agar's prescription of working with "a little bit of data, and a lot of right brain" (Agar, 1991: 194). Sometimes the process took us beyond the topic of the segment. Sometimes it took us deeper into the topic.

Threads and Patterns in the Data;
Mapping the Data Landscape

Agar's second analysis alternative is the analogy of the "multi-threaded DNA laid on the text so you [can] look at the patterns that each thread reveal[s], and the patterns among the patterns"(Agar, 1991: 193). I argue that, because The Ethnograph lets you see your code words embedded in your

data file, it directly facilitates this type of analysis. But before I address this I will offer two similar analogies: topographical maps and ad hoc maps.

A topographical map is a way of coding the landscape so that it shows you the physical features of the landscape. It gives you a very different picture of the physical landscape compared to a standard road map. It shows you the hills and valleys, forests and clearings, and other features and details of the landscape in relationship to each other. This makes it easier for you to navigate through unfamiliar territory, especially off the roads. In a similar manner your "codes" can highlight features and details of your data landscape.
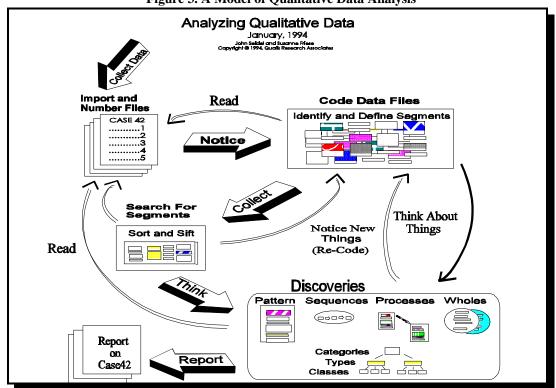
The display of code words embedded in the data file (which The Ethnograph does) produces something resembling a "topographical" map of your data. Just as a real topographical map can help you discover and chart a path through the countryside, your codes can help you discover and chart patterns through the "landscape" of your data. These patterns are not reducible to code words, and are not discoverable from a simple examination of collections of coded segments. Yet these patterns can only be discovered because of the way in which you coded your data.

An ad hoc map is the kind of map that you draw to tell people how to get to your house. When you draw this map you highlight (i.e., code) certain features of the landscape as reference points. For example, you might emphasize major intersections, stop lights, stores, etc. There are many intersections, stop lights, and stores in the area, but a particular combination of them mark the path to your house. In order to draw the map you have to know some general things about intersections, stop lights, and stores. But this general knowledge does not reveal the path to your house. Knowing and describing the path requires a knowledge of specific intersections, stoplights, and stores. Thus, describing the path depends on coded features of the landscape, but the path is not reducible to the coded features, nor is it revealed by studying collections of those features of the landscape.

A practical example from my own work illustrates the "threading" or "map" metaphors. It comes from another analysis session on the second stage labor project.   One day my colleagues and I focused on a data fragment where the nurse displayed the "three push rule" to the laboring woman. The plan was to intensively analyze this small piece of data.

While two of us attended to this data fragment the third member of the team drifted away from the discussion and started looking at the fully coded transcript from which the segment came. She noticed the co-occurrence of a "praise" utterance with the "three push rule" display. She also noticed that this came at the end of a uterine contraction. Going back through many pages of the transcript she noticed the absence of a "praise" utterance during all the previous contractions. After she brought this to our attention we followed the patterns backwards to another pivotal event, and then forward to the "three push rule" display. In this way we discovered a new phenomenon, a "progress crisis," which cut across, and transcended, the coded segments on the transcript.

This discovery depended on the fact that we had coded our transcript in a particular way, but the discovery was not reducible to the codes, nor could it have been derived by simply inspecting collections of coded segments. Because of the way we had coded the data, the features of the data landscape were highlighted on the transcript, the "threads" were visible. Through the combination of: 1) focused attention, 2) intensive analysis of a small part of the data, and 3) the ability to see the how several "threads" or "features" of the data came together over several pages in the transcript, we were able to make the "progress crisis" discovery.

Figure 3. A Model of Qualitative Data Analysis

A Complex Model of the QDA Process

The diagram in **Figure 3** is a model of the features of the QDA process, and how a computer program fits into this process. It incorporates and builds on the basic *Noticing, Collecting, Thinking* process that I have been discussing. This diagram consists of four parts:

1. Arrows ⟹ which represent the basic Notice, Collect, Think process described in the previous section.

2. Three Boxes ☐ which represent the three basic processes of The Ethnograph: Import and Number, Code a Data File, and Search for Coded Segments.

3. A large box with rounded corners ⬭ that represents analytic "discoveries."

4. Light curving lines with arrows ⬭. These represent the iterative and recursive aspects of QDA.

The Processes of Noticing, Collecting, and Thinking about Things

The large arrows ⟹ represent the basic processes of qualitative data analysis:

- **Collect** Data.
- **Notice** interesting things in your data.

- **Collect** sets of those interesting things.
- **Think** about those interesting things.
- **Write** a report about those things.

Three of these (Notice, Collect, and Think) were discussed in the first part of this appendix. The other two arrows represent the entry and exit points of the process (Collect Data and Write a Report).

While the placement of these arrows suggests that the process is progressive and linear, the diagram preserves the nonlinear, iterative, and recursive aspects of the process as discussed in the previous section.

The Basic Procedures of The Ethnograph

The boxes ⬜ represent the basic procedures of The Ethnograph (Import and Number Files, Code Data Files, and Search for Coded Segments). For an overview of these procedures see Chapter 4: Quick Tour.

1. **Import and Number Data Files:** This procedure takes data files that you have created (using either your word processor, or the Ethnograph Editor), and transforms them into "numbered" data files that The Ethnograph can process. These are the basic data records that you read and analyze.

    NOTE: Later, when I talk about going back to Import and Number, I mean going back to the data records you previously created when you ran the Import and Number procedure, not the Import and Number process itself.

2. **Code Data Files:** This procedure facilitates the process of identifying and naming interesting things in your data files. The differently shaped and shaded boxes inside of this figure represent the various types of things you might notice and code in your data. At this point you are still dealing with an undifferentiated mass of coded data.

3. **Search for Coded Segments**: The next step is to bring order to your data. This means disassembling and reassembling the data set based on your coding scheme. We also call this process "sorting and sifting" your data. This makes it easier for you to closely examine, and compare and contrast, things that you notice in your data. This process is represented by the orderly display of shaded boxes in the figure.

The Swirls and Eddies of the Analysis Process

The light curving arrows ⌒ represent **the swirls and eddies** of qualitative data analysis. Analysis does not just happen. It evolves and develops in an iterative and recursive fashion. As the analysis develops you learn to think differently about the data you have already collected. As you progress through the various steps in the process you are constantly returning to previous steps.

Coding Influences Coding and Analysis

The initial coding work that you do (represented by the Code Data Files box) helps you notice new things in your data. Notice the arrow that goes from this box back to the Import and Number Files box. This arrow is labeled *Read*. This means that you need to read your original data files again. The act of coding changes both the original data and your relationship to that data. As you start to code you will discover other things to notice and code.

Searching Influences Coding and Analysis

The initial sorting and sifting has three effects:

1. It leads to revisions in your coding scheme. This is represented by the arrow that goes from the box labeled Search for Segments to the box labeled Code Data Files.

2. It helps you notice new things in your data. This is represented by the arrow going from the Search for Segments box to the Import and Number Files box.

3. It facilitates the process of thinking and making discoveries.

The Emergence of Discoveries

Discoveries emerge in many ways and can take many forms.

- Sometimes discoveries emerge from the sorting and sifting process
- Sometimes discoveries emerge from simply examining the coded transcript.

This latter path is represented by the dark arrow going from the Code Data Files Box to the Discoveries box.

Some types of discoveries are represented in the Discoveries box ▭ at the lower right hand corner of the diagram. Discoveries can be patterns, sequences, processes, wholes, classes, types, and categories.

Discoveries Influence Coding and Analysis

The initial discoveries that you make are preliminary and provisional. They have two effects on data analysis.

1. They help you notice new things in your data. This is represented by the arrow going from the Discover Box to the Import and Number Files box.

2. They suggest revisions to your existing code map. This is represented by the arrow going from the Discoveries box to the Code Data Files box.

Codes as Heuristic Tools

This essay has been premised on the idea that code words in QDA are primarily heuristic tools. Earlier in this essay I alluded to an objectivist-heuristic continuum for understanding code words. While this distinction has been described elsewhere (Kelle and Seidel, 1995), I would like to make a few remarks about this distinction in the conclusion to this essay.

First, following on a previous discussion (Kelle and Seidel, 1995) I want to make it clear that I am not talking about code words per se. Code words are not inherently objectivist or heuristic. Rather, these are terms that describe how we think about, and make use of, code words in QDA.

Second, I am not talking about an either/or distinction. In any given research project some code words might be more "objective" and others more "heuristic." Further, some code words might be used for both purposes. Depending on your analytic style and purpose you might gravitate toward one or the other use of code words.

But I do think that the tradition in QDA is primarily to treat code words as heuristic tools rather than objective representations of facts. A tendency to treat code words "objectively" is, at best, problematic.

Objectivist Codes

An objectivist approach treats code words as "condensed representation of the facts described in the data" (Seidel and Kelle, 1995). Given this assumption, code words can be treated as surrogates for the text, and the analysis can focus on the codes instead of the text itself. You can then emulate traditional distributional analysis and hypothesis testing for qualitative data. But first you must be able to trust your code words.

To trust a code word you need: 1) to guarantee that *every time* you use a code word to identify a segment of text that segment is an *unambiguous* instance of what that code word represents, 2) to guarantee that you applied that code word to the text *consistently* in the traditional sense of the concept of reliability, and 3) to guarantee that you have identified *every* instance of what the code represents.

If the above conditions are met, then: 1) the codes are adequate surrogates for the text they identify, 2) the text is reducible to the codes, and 3) it is appropriate to analyze relationships among codes. If you fall short of meeting these conditions then an analysis of relationships among code words is risky business. I have identified some of these risks in an earlier work (Seidel, 1991).

Heuristic Codes

In a heuristic approach, code words are primarily flags or signposts that point to things in the data. The role of code words is to help you collect the things you have noticed so you can subject them to further analysis. Heuristic codes help you reorganize the data and give you different views of the data. They facilitate the discovery of things, and they help you open up the data to further intensive analysis and inspection.

The burdens placed on heuristic codes are much less than those placed on objective codes. In a heuristic approach code words more or less represent the things you have noticed. You have no assurance that the things you have coded are always the same type of thing, nor that you have captured every possible instance of that thing in your coding of the data. This does not absolve you of the responsibility to refine and develop your coding scheme and your analysis of the data. Nor does it excuse you from looking for "counter examples" and "confirming examples" in the data. The heuristic approach does say that coding the data is never enough. It is the beginning of a process that requires you to work deeper and deeper into your data.

Further, heuristic code words change and evolve as the analysis develops. The way you use the same code word changes over time. Text coded at time one is not necessarily equivalent with text coded at time two. Finally, heuristic code words change and transform the researcher who, in turn, changes and transforms the code words as the analysis proceeds.

Somewhere in the Middle.

In practice most code words in QDA fall somewhere in between pure objectivist and pure heuristic coding. The key words here are "in between." In pure objectivist coding you can blindly trust your code words. But in order to do this you must place some heavy burdens and expectations your code words. If your code words cannot carry these burdens, and meet these expectations, then any analysis premised on an objectivist treatment of code words is problematic. Yet, even in heuristic coding you need some level of confidence in your code words.

Finally, I am not saying that the analysis of codes and relationships among codes is never useful. It can be useful in a heuristic sense. But it always has to be tempered with skepticism and doubt about the analysis. The ultimate appeal is always to the text itself. (See the example in **Appendix C, pages C-20 to C-22**).

To paraphrase Shakespeare, the answers we look for are not in the codes, but in ourselves and our data.

Agar, Michael (1991) "The right brain strikes back", in *Using Computers in Qualitative Research*, N. Fielding and R. Lee, eds., Newbury Park: Sage Publications, 181-194.

Charmaz, Kathy (1983) "The grounded theory method: An explication and interpretation", in *Contemporary field Research: A Collection of Readings*, Robert M. Emerson, ed., Boston: Little, Brown and Company, 109-128.

Corbin, Juliet and Anselm Strauss (1990) *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*, Newbury Park, CA: Sage Publications.

Freidson, Elliot (1975) *Doctoring together: A Study of Professional Social Control*, Chicago: University of Chicago Press.

Jorgensen, Danny L. (1989) *Participant Observation: A Methodology for Human Studies*, Newbury Park, CA: Sage Publications.

Schneider, Joseph W. and Peter Conrad (1983) *Having Epilepsy: The Experience and Control of Illness*, Philadelphia: Temple University Press.

Seidel, John (1991) "Method and Madness in the Application of Computer Technology to Qualitative Data Analysis", in *Using Computers in Qualitative Research*, N. Fielding and R. Lee, eds., Newbury Park: Sage Publications, 107-116.

Seidel, John and Klaus Udo Kelle, and (1995) "Different Functions of Coding in the Analysis of Data." in K.U.Kelle ed., *Computer Aided Qualitative Data Analysis: Theory, Methods, and Practice*. Thousand Oaks, CA: Sage Publications.

Spradley, James P. (1979) *The Ethnographic Interview*, New York: Holt Rinehart, and Wilson.

Wiseman, Jacqueline P. (1979) *Stations of the Lost: The Treatment of Skid Row Alcoholics,* Chicago: The University of Chicago Press.