QSW Qualitative Social Work
Vol. 1(4): 473-487 Copyright ©2002 Sage Publications London, Thousand Oaks, CA and New Delhi
1473-3250[200212]1:4;473-487;029514

TECHNICAL APPLICATIONS

Musings of a Skeptical Software Junkie and the HyperRESEARCH™ Fix

Karen M. Staller
University of Michigan, USA

ABSTRACT
This article considers some of the hazards and benefits of using computer-assisted software to analyze qualitative empirical evidence. Specifically, I discuss HyperRESEARCH, one software package currently on the market and available to both Macintosh and PC users. This article is not intended as a substitute for the user's manual nor to be an exhaustive discussion of the software's features. Instead I attempt to illustrate some of HyperRESEARCH's unique characteristics, suggest how qualitative researchers might think about using them and provide some basic practical advice.

KEY WORDS:
Computer-assisted qualitative analysis software
HyperRESEARCH
qualitative analysis
qualitative methods

'SOME DO, SOME DON'T': ADVICE FROM A SAGE SPIRIT[1]

I never met Elliot Liebow, but I know I would have liked him, because of the humility and humanity in his writing. His last work, *Tell Them Who I Am: The Lives of Homeless Women* (1995), is a participant observer study on homeless women. It is also a labor of love written by a distinguished scholar diagnosed with terminal cancer, using a bit of his remaining time to do what came naturally (Liebow died in September 1994). A lifetime of wisdom is reflected in every sentence.

Why begin a manuscript on computer software with this Liebow tribute? First, he is a role model and scholar to be reckoned with. Second, because his work serves as admirable warning about why we might wish to proceed slowly and skeptically before jumping on the software bandwagon. Buried in Liebow's 'research appendix' in *Tell Them Who I am* – following a painstaking but illuminating discussion of the process of manual qualitative analysis – is the ominous footnote four which reads:

> An anonymous reviewer describes this 'fussing' with my notes as 'dinosaur-like' in view of all the computer software available for dealing with qualitative data. My problem, however, was not retrieval or even organization of my notes but rather seeing how they relate to one another and integrating them, and I did not see how to make the software do this for me. Sometimes we do better by sticking to what we think we know. (Liebow, 1995: 324)

I do not disagree with him. There are serious warnings embedded in his appendix about why *not* to use software, and they should be remembered as admonishments in his memory. So that is where I start, when asked by doctoral students or colleagues who are out there trawling for the 'best' software, I ask them why they want to use software at all (see Drisko, 1997; Weitzman, 2000). A typical response is a surprised look and a vague reference to rigor as though software, by definition, increases rigor – it does not.

While Liebow was not attempting to critique the dangers of software, his writing suggests at least several points worthy of note. First, the observation that spawned footnote four was that 'however tedious' the manual procedure of moving from note card extractions of material to full pages of original text, it gave him 'an additional and valuable familiarity with my notes' (Liebow, 1995: 324). So, first and foremost, qualitative researchers must never lose familiarity with the text during data analysis. With software, because you can ask a machine to manipulate isolated and manageable pieces of text, these parts can take on a life of their own and may do so at the expense of the greater context from which they are drawn. This must not happen, software or not.

Second, in footnote four Liebow worries software will not help with how to 'relate [concepts] to one another and integrating them' (Liebow, 1995:

324). Here are two important ideas linked by the Boolean operator 'and' (more on Boolean logic to follow). We must take note of both sides of his equation. The first part is that software will not discover connections for you. While some may argue that because software has unlimited capacity to supply the researchers with proximal information (e.g. Do concepts overlap, appear sequentially or concurrently? How close or far apart are they?, etc.) that this somehow proves relationships. Yet the researcher is responsible both for the mechanical decisions that lead to marking text which created the proximal relationships in the first place and for asking the questions about the existence of the relationships. The analytic work still belongs to the researcher.

The second half of Liebow's warning involves 'integrating' the material. Data reduction techniques break text into smaller pieces but at some point those pieces must be synthesized into an integrated whole, and the final written product must reflect that whole. Organizing and managing data should not be confused with the techniques a researcher uses to make sense of it. Data re-configuration – removing it from its original place in a text and linking it, or relating it to other ideas – requires an analytical framework. Computers do not create the framework for you. Software will never substitute for a research question, project design, interpretation, or thoughtful reporting. In short, Liebow is right that software will not do the work for you; nonetheless, there are some responses to the explicit and implicit concerns raised by Liebow while reflecting on his struggles during analysis.

Managing Volume

In his research appendix, Liebow attributes his biggest problem to having collected too many notes over too long a period of time. Of this dilemma he writes, 'It is quite possible that in taking many notes over a long period of time, I bit off more than I could chew' (Liebow, 1995: 324) and that part of the problem, 'is traceable to the fact of too many notes' (Liebow, 1995: 325). Managing unlimited amounts of empirical evidence is something computers do extremely well. This begs the question about whether volume is, in and of itself, a good thing. More is not always better. Nonetheless software can help manage volume no matter how voluminous.

Aiding Memory

Equally perplexing to Liebow as the volume of his notes was his inability to grasp their meaning all at once. He wrote:

> I had the feeling, while writing, that I was dealing with them piecemeal; that because of the volume, I could pick up only one corner of the notes at a time. I felt I could, perhaps, connect one piece with another, but I never had the feeling that I could embrace them all at once, or that, altogether, they made special sense of their own. (Liebow, 1995: 324)

The human brain's capacity to move material about while holding on to the overall picture is limited. Computers can serve as memory support. While you pick up 'one of corner of the notes' you can keep the rest in mind with the machine's help – this is true if and only if you are firmly in control of what you are asking the software to do for you.

Comparing Things

Liebow finds an answer to his quandary of making sense out of his notes in the work of Charles Darwin who argued 'all observation must be for or against some view if is it to be of any use' (Liebow, 1995: 325). Liebow (1995) writes 'I needed something to be for or against' (p. 325). He illustrates the process of his intellectual discovery: 'How do women survive the inhuman conditions that confront them? After I wrote that paragraph, it occurred to me that most situations, experiences, and processes could be seen as working for or against the women's survival and their humanity . . . ' (p. 325). Using this comparative strategy, Liebow found a way to structure his entire study. Comparing things is useful not only because it helps locate the stories which are interesting to tell, but also because it addresses Liebow's earlier dilemma of picking up 'one corner of his notes' at a time and losing the rest. Comparing is a process of constantly contrasting. Examining similarity and difference is a skill we develop from childhood. It is what we do naturally.

This notion leads me to HyperRESEARCH. If there is one computer assisted qualitative software on the market at the moment that specializes in forcing researchers to consider what they are for and against, it is this one. HyperRESEARCH's primary unit of analysis is the *case*, not the *document*. It forces researchers to conceptualize and structure their entire analysis around the very idea of being for and against.

INTRODUCTION TO HyperRESEARCH

For and Against: A Comparative Advantage

Perhaps the single biggest difference between HyperRESEARCH and other software programs currently on the market is its case-based design. Other programs permit you to open up a data file (e.g. interview transcript or other document) and start coding directly. By default, the unit of analysis is the document. HyperRESEARCH forces the researcher to start with a different unit of analysis called the case. Lest there be any doubt about it, the software opens directly to a case card screen, so you are confronted immediately with the case-based conceptual framework.

The largest organizational unit in HyperRESEARCH is the study.[2] Obviously, you can have as many different studies in progress as you can handle. However, you must start each study with at least one case. Within a study you

may open as many cases as desired. So each research project must be organized around the idea of the case.

Each individual case card may be linked to an unlimited number of sources (e.g. data, empirical evidence), which can be of three types: text, image, or video/audio.[3] Furthermore any single source may be linked to any number of cases. There is no limitation on linking sources and case cards.

So what does that mean in English? Say your social work practice involves art therapy with children over time. You see 20 children in 15 sessions. For each session you have a transcription of the interview with the child, field notes, a photograph of the final art product, and videotape of the child creating it. Unlike other software on the market where you could open up a text file and start coding, in HyperRESEARCH you link these sources to one or more cases first, which means that you must decide what your biggest unit of comparing and contrasting will be. It is best, at this point, to consider your research question and how you plan to use theory in your study, otherwise you can get lost in the mechanical process of using the software without direction. HyperRESEARCH, of course, will not design your study for you.

If your research question focuses on children you might create a case card for each child. The first child's case card would be linked to all the text, photos, and videos you have collected over 15 weeks of sessions with that child. Similarly, the second case would be linked to all the empirical evidence on child two. This would proceed until all 20 children had their own case. Of course, you can add new cases to a study whenever you want. So if you acquire a new client after you have started a study, it is easy to add another.

On the other hand, perhaps you would rather ask a research question about the 15-session intervention. In this situation your first case card might be called Session One and it would be linked to the first session empirical evidence for all 20 children. In other words, all 20 first interview transcripts, all 20 first artworks, all 20 first videotapes, etc. You would proceed until you had 15 case cards, one for each session.

Finally, you may want to ask some entirely different type of question that would suggest setting up case cards based on conceptual categories. Perhaps you have some theoretical plan for structuring your study. For example, you are studying external support networks so you want to create case cards based on relationships such as Parent, Teacher, Friend, Peer, etc. If a child talks about his parent in session one, and his teacher and parent in session two; then the first case card (Parent) would be linked to the first transcript but the second transcript would be linked to two cases (Parent and Teacher).

For me, an early mistake was to think too linearly and link one case card with each new data source. In fact, you can structure your study with unlimited sources linked to unlimited cases. I first learned this lesson in a study I did of social work dissertation acknowledgement pages (Staller, 2001). I originally

created a case card for each student and linked it to his or her acknowledgement page text. The unit of analysis was the student. However, during analysis it became clear that certain faculty members, with unique mentoring and teaching characteristics, were particularly important to the program (as recorded in acknowledgment pages anyway). So I created a case card for each of these faculty members and linked it to all references to that person, as well as references to other faculty mentioned in conjunction with the primary mentor. In short, the faculty member became the unit of analysis, with his or her case linked to many different student acknowledgement pages.

The importance of the case becomes evident as you proceed to analysis. However, the clearer you are about where you want to go at the outset, the more efficiently and effectively you will be able to utilize this software design. Of course, cases can be whatever conceptual unit you want: individuals, concepts, organizations, political entities, states, countries, service providers, etc.

I have talked about 'linking' cases with sources but what do I mean by *linking*? The link between document and case is created by the code. As the researcher applies codes to the source, they appear visually in the margins of the source document, but more significantly the code appears on the case card. So each case card will begin to accumulate a long list of codes, which are hyperlinked to assorted texts, images, audio, and video sources. If you click on any code on the list it will automatically open the original source document and highlighted code material.

A final note about cases before moving on to coding: I always create an additional case card for my study literature. I link this case card to all the articles, books, news accounts, or other secondary materials that I have read on my study topic and might use when writing the final research report. This way, I can code my reading using the same codes as my study. In analysis this allows me to generate reports that contain literature pertaining to my study concepts as well as my own empirical evidence.

Making Links: The Coding Process in HyperRESEARCH

As noted above, the unit that links the *source* (text, image or audio/video) to the *case* is the *code*. At this point it is important to recall any and all of your previously acquired knowledge on the perils of coding. HyperRESEARCH does not eliminate any of them, in fact, it increases the importance of code boundaries.

Coding, of course, involves attaching a descriptive label to a segment of text, an area of a photograph, or a segment of audio or videotape. *Coding* is defined as a 'procedure that desegregates the data, breaks it down into manageable segments and identifies or names those segments' (Schwandt, 1997: 16). Good coding form includes writing a definition describing the code that is clear enough to establish the boundaries of the code concept itself. HyperRESEARCH has a floating (moveable) code editor window which is used to

create and apply codes to source documents. The window also has a code annotation feature so you can keep track of the evolving definition of the code, including exemplars. The annotation appears whenever you highlight the code on the code editor list.

The mechanical process of coding is deceptively easy. For text documents, you highlight the text you want coded and click on 'apply code' (or 'new code' if you are creating a new one) in the code editor window. For image files, you create a box around the area of the image you want to code and click on 'apply code.' For video or audio files you isolate a segment of tape by dragging the cursor along the play bar, thus highlighting the segment you want coded.

The real difficulty and the danger of coding in HyperRESEARCH (as with other software programs) is twofold. First you must decide where to start and stop the highlighted segment. What text is included in the code and what is excluded? These decisions become critical during analysis because the boundaries of the code become important. These boundaries, of course, are constructed by the researcher, as are the codes themselves. So for example, in a study I did of a police unit, I had extensive transcriptions of conversations that were interspersed with stories told by the officers (Staller, 2002). I wanted to separate the stories from general conversation so I created a 'narrative' code. However, I had to decide where the story started (and other conversation stopped) and where the story ended (with conversation resuming). It sounds easy but those boundaries are negotiated in general conversation.

Second you must decide how broadly or narrowly you plan to code at the beginning of the process. I have learned to start broadly and to refine my ideas later in the process. For example in the acknowledgement page study, I coded every social service agency mentioned by name until I realized I had hundreds of agencies listed and did not know what to do with them all. I began to convert my codes to broader and more general categories (e.g. hospitals, schools, senior centers, etc). Recently, when I referred to my narrow coding as a 'beginner's error' in a presentation to doctoral students, I was soundly chastised by those who felt that extremely detailed coding was the only way to rigorously and systematically explore empirical evidence and to build ideas. So I have modified my stance, but I would still urge caution. Just do not mistake the busy work of detailed coding for forward progress if you do not know why you are doing it.

HyperRESEARCH also has an autocoding feature, which permits the researcher to ask the software to search lots of source text for a specific word or phrase (or words within a specified proximity to each other) and to add a code automatically whenever it sees the specified text. It is possible to ask the software to apply the code to the selected word or words. However, it also possible to attach the code to any number of characters, words, or paragraphs before or after the selected word (or words). For example, I once asked HyperRESEARCH to

search for the word 'because' in transcripts of adolescent girl sexual abuse survivors and to include the 10 words before and the 10 words after the word 'because' in the coded segment. My thought was that 'because' would be important in the girls' speech linking cause and effect. It turns out that teenage girls say 'mmm, because, you know, just because' fairly regularly, which leads to an important proviso: never autocode text without immediately going back and checking to see what you have gotten in the process. It is easy to delete or rename codes – on an individual case basis or globally – as needed. HyperRESEARCH is very protective of the foolish error-maker. If you use the delete command a window automatically appears that asks if you are *sure* you want to proceed with the deletion. Thus, accidentally deleting your work takes concerted effort.

Whatever the medium being coded, I warn my students that it is extremely easy to get swept up in the mechanical process of coding. It is fun and you feel like you are making progress until you find yourself with a very long list of codes and nothing to do with them. This is a good place to invoke Liebow's warning that the software will not tell you how these coded segments relate to one and other or what they mean without your direction (Liebow, 1995: 324). Asking HyperRESEARCH about the relationship between codes happens as you generate reports (see generating reports section below).

Boolean Logic, Proximity, and Coding: Value added or lost?

Perhaps what makes computer software most different from the manual process of finding themes is the additional complexity created by the ability to use Boolean logic and proximity locators in the analysis. We should be both awed and wary of this capacity and ask continually: is this a useful addition to the way qualitative researchers go about the business of analysis or not?

HyperRESEARCH software permits examining possible relationships endlessly by selecting cases that comply with certain conditions. These relationships can be specified with great complexity. You may use Boolean connectors (and, or, not) and proximity functions (equals, includes, overlaps) or any combination of both to make your selections.

Start with the notion of highlighting segments of text and Boolean operators: *and, or.* If you have highlighted two codes A and B, Boolean logic permits you to work with 'A and B.' This gives you all the text where A and B overlap (or the cases that contain this condition). You may ask to work with 'A or B' in which case you get all the text for both codes (or the cases that have one or the other code). You may also use proximity locators or functions: equals, includes, overlaps. Thus you can specify that you want to work with cases in which Code A overlaps with (Code B if it equals Code C). Just to add to the mind-numbing complexity, you may use any combination of Boolean operators coupled with proximity locators. Furthermore, you can introduce an unlimited number of codes to your equations.

Note how important your initial code highlighting becomes at this stage. Where you begin and end your code highlight now determines how your codes will interact. The importance of these features becomes clearer as you generate reports and examine results (see further discussion in section below on generating reports).

At this point, it is critical to wonder if this complexity adds value to qualitative analysis. Can we really learn anything of importance in these detailed relationships? Is this technical wizardry devoid of meaning? I would suggest that this is where software might alter the very essence of the research process as it begins to impose a formal structure of analysis on the researcher trying to make sense of empirical evidence.

For example, I used proximity in my analysis of police officer conversations and narratives; however, more recently in a study on adolescent girls sexual abuse survivors, it took me a while to discover this feature was a hindrance rather than a help. In this analysis I started by looking at the detailed relationships between concepts within the girls' stories. In doing so I lost the essence of their core message, which turned to be infinitely more important. I retreated from a deconstructive approach to analysis and resorted to a more holistic approach. Although I still used HyperRESEARCH to manage my thoughts and observations, I did it through analytical notes and summaries of the girls' statements rather than by micro-analyzing the text of their talk.

Generating Reports: The Power of Activating, Suppressing, and Linking

Perhaps the most conceptually difficult and challenging feature of using HyperRESEARCH happens in the process of generating reports. Reports permit you to sort and search your data and to examine what you have. Of course, Liebow is whispering a reminder that none of this will tell you how things relate (Liebow, 1995: 324). However, HyperRESEARCH gives you unlimited ability to play with how things might relate and greatly speeds up the process of experimentation.

The first step in understanding the power of HyperRESEARCH reports is to understand that you can activate and suppress any sub-sample of cases and/or codes. Cases may be selected based on case name or by specified criteria. The specified criteria may be organized by codes or by code combinations using Boolean logic and proximity locators as described above. Furthermore, codes themselves may be activated or temporarily suppressed. Again, this can happen by code name or by specified criteria, so reports can be generated for subsets of cases and/or codes. The report itself can be presented grouped either by case or by code. Finally, the reports are hyperlinked to the original source, if there is a segment of text in the report and you would like to hop back to the original source document from which it was drawn, all you have to do is click on the link. This gives the researcher the opportunity to heed Liebow's warning to

become familiar with the text by moving back and forth between selected segments and the whole document (Liebow, 1995: 324). With HyperRESEARCH you can do it electronically with just a click of the mouse.

So how does this work? First, concentrate on the notion of your cases. Any subset of cases may be activated for analysis (thus the others are temporarily suppressed). For example, suppose you are studying those 20 children in art therapy and you have coded for the child's gender as well as concepts you are observing in their work and conversation. You want to compare progress by gender. You could select cases based on the criteria 'girl and positive progress' assuming you have coded for gender and positive progress. If 8 of your cases contain both of these code features then your case card tool bar will read 8 of 20, indicating you are now working with the subset of cases that contain both these features. You can run a report on this sub-sample. Next you could select cases based on the specified criteria 'boy and positive progress.' Perhaps you get 9 cases. You can now run a report on the boys separately, which permits you to examine what factors may be present or absent in the positive progress of your subjects by gender. Of course in my scenario you have three cases left out of the analysis, either because you failed to code for gender or because there was no positive progress code. So when you asked for cases with 'gender and positive progress' they were not selected.

However, this is not the end of the flexibility. Not only can you activate and suppress cases but you may use the same logic to activate and suppress codes on case cards. Perhaps you have created 210 different codes for the therapy study. In the analysis above your report for each case (8 girls and 9 boys) will provide information on all 210 codes. However, suppose you want to work with a subset of codes. Perhaps you would like to look only at verbal expression. You have created 30 different verbal expression codes out of the 210 total codes. By activating all the verbal expression codes, only those will appear on the activated case cards. So if case one contains 15 different verbal expression codes out of 22 codes on the card, your card will read 15 out of 22 selected. You will only see the 15 selected verbal expression codes. If your second case contains 6 verbal expression codes but 199 total codes, your case card will read 6 of 199 selected and you will only see the 6 verbal expression codes. Now you can generate a report that will provide information on only these specified verbal expression codes.

If you combine the case and code activation features described above you can study readily how girls who are making positive progress express themselves verbally compared to how boys who are making positive progress express themselves verbally. Obviously, you can activate and suppress any combination of cases and any combination of codes. At this point it seems worth remembering Liebow's discovery that it is useful to be 'for and against' things in analysis (Liebow, 1995: 325). Here is where HyperRESEARCH permits unlimited

capacity to construct 'for and against' scenarios and examine what is happening in the empirical evidence. In short, you can ask endless and complex comparative questions of your empirical evidence.

Finally, reports can be generated grouped either by case or by code. Thus, if you have 8 cases activated and you ask for a report by case, you will get 8 (electronic) pages. Each page will contain information on the activated codes for that case. On the other hand, you can also ask for the report to be sorted by code. If the total activated codes in your analysis is 50 then you will get 50 (electronic) pages. The first will be on code one and contain information on the cases that utilize code one. The second page of the report will be on code two and contain information on all the cases that utilized code two and so on. Thus you can study your results either by case or by code. Of course, which format you choose will depend on what question you are asking.

In summary, you have enormous power and control over examining combinations of isolated and grouped material. While this can be exhilarating, it is critical to stay in conceptual control of what you are doing. Remember to be analytically and conceptually clear about what you are asking the computer to do and why.

Hypothesis Testing or Theory Building

A second major difference between HyperRESEARCH and other software programs is its 'hypothesis testing' capacity. The very notion of hypothesis testing offended my qualitative sensibilities so I avoided using this feature for a year before I discovered how useful it could be. In the privacy of my own office I simply think of it as the theory building and evaluating tool.

There are two important steps for using the hypothesis tester. The first is to construct useful if/then statements. In creating the *if* clause you may include or exclude any combination of codes and/or use proximity functions (equals, includes, overlaps). The *then* clause tells HyperRESEARCH to take an action including adding a new or pre-existing code, deleting a code, or adding a 'goal.' So, for example, you can say 'if Code A and (Code B overlaps Code C) then add Code Z.' This will tell HyperRESEARCH to search each case card to see if the 'if' condition is met. If HyperRESEARCH finds these conditions it will automatically insert a new code 'Z.' Note that Z is not directly linked to source text the way A, B, and C are. Newly added Z is a conceptual code that is inserted electronically only when certain specified conditions are met. Yet this abstract code can now be used in analysis as though it was a directly linked to text.

For example, perhaps codes A, B, and C mark specific verbal expressions used by children during art therapy. However, together those expressions would lead you to conclude the child is building self-esteem. So you tell HyperRESEARCH to insert a self-esteem code (Z). Now you have an abstract self-esteem code that is built upon a series of concrete observations. You may now

use self-esteem as you would any of the direct hyperlinked codes in future analysis. It is important to note that HyperRESEARCH will look for these conditions within each activated case. It should be clearer, now, why your original design structure is so important – if you have originally designed your study using 15 cases by 'Session' rather than 20 by 'Child' it would be harder (although not impossible) to do this particular analysis.

In addition to adding or deleting codes as part of the 'then' action, you may also ask HyperRESEARCH to signify that a 'goal' was achieved. The researcher can create a list of as many if/then statements as she wants and if all of them are met the computer will report that the goal was obtained for that case, which is a way to test theoretical propositions. Take note, once again, of the importance of the case design. HyperRESEARCH will test the propositions you specify by case card, so if you plan to use this feature you must be sure that your cases are constructed correctly for testing or generating theory from the beginning.

CONCLUSION

My initial partnership with HyperRESEARCH was a bit of a forced marriage. As a confirmed Macintosh user my options were limited, and for Mac devotees the options are even more limited today since the latest versions of NVivo have abandoned the Mac platform. HyperRESEARCH continues to be available for both Mac and PC users. In fact, I have shared files successfully across platforms with colleagues. Let me hasten to say I receive no financial kickbacks from the company nor would I characterize myself as a HyperRESEARCH zealot. However, I have used it and liked it. I continue to discover new ways to make it work for me. It provides conceptual and intellectual challenges.

In particular, I find its window environment, with a tool bar across the top, and its floating (and resizable) windows for the case card, source document, analytical notes, code editor, hypotheses tester, etc., useful. You can move these windows around your screen for convenience. Furthermore, because HyperRESEARCH looks and acts a lot like Microsoft Word, it is easy to master basic functions. For example, the cut, copy, paste, save, save as, etc., commands operate exactly as a Microsoft user would expect. For that reason it is also a nice tool to use with students. Of course, the difficulty with HyperRESEARCH is not in learning the simple mechanical tasks, it is in exerting analytical control. The HyperRESEARCH online experts are very responsive to user problems and helpful in solving them. Finally, I am attracted to HyperRESEARCH – in keeping with my natural social worker inclination to champion the underdog – because it has quietly continued to improve without the fanfare of some of its more aggressively marketed competitors which arguably have worked as hard at name recognition as on the quality of the software itself. HyperRESEARCH

version 2.5 will be available as a free download for registered 2.0 or 2.03 users. The latest version is very much improved so make sure it is the one you are using.

I am fascinated by the competition between developers, which is currently both fierce and in its infancy. For example, in the public domain, the Centers for Disease Control and Prevention (CDC) has developed a free downloadable software called AnSWR (see http://www.cdc.gov/hiv/software.htm). Given the kind of large-scale, multi-site research CDC encourages, it is not surprising its developers are solving questions of inter-rater reliability, the interface between qualitative and quantitative analysis tools, levels of security and restricted access, confidentiality protections, etc. On the commercial side, the worldwide competition includes Australia (NUD*IST, NVivo); Germany (Atlas.ti, Winmax); and the USA (Ethnograph, HyperRESEARCH). The developers come out of different disciplinary backgrounds (e.g. psychology, anthropology), which may influence the design (see Miles and Huberman, 1994). At some point, I suspect the field will prune itself. In the meantime, it is fascinating to watch where they are all going, and what design problems they are tackling. The design race is critical because it will shape the tools available in the future and perhaps influence the way qualitative researchers do business.

This leads me back to Liebow's footnote four, 'some do, some don't,' (Liebow, 1995: 320, 324) and my own note one in which I ponder the implications of using software and whether these tools will begin to alter the research endeavor itself by influencing the way we think about and structure our analyses. After reading this manuscript, a sympathetic colleague recently asked me whether HyperRESEARCH is useful at the exploratory stage or only when the direction of a study is already determined. The easy answer is that you can, of course, use HyperRESEARCH as an exploratory tool. However, therein lies the essence of my concern. The more experienced one becomes with the software, the more likely it is to influence the way we establish direction even in the preliminary stages. To the extent that this happens, the software is altering the essence of the inquiry itself.

Acknowledgement

The author thanks Dr Larry Root for his helpful suggestions on a draft of this manuscript.

Notes

1 'Some do, some don't' is a Liebow reminder that exceptions 'may not be exceptions at all. They may be a different way of doing things, a different way of behaving, and instead of 'proving the rule,' they may be evidence of a different rule, a different pattern' (Liebow, 1995: 320). Some use qualitative software all the time, some never use it. I have used HyperRESEARCH for some projects but not for others. Frankly,

I am more concerned with method than technology. However, I confess I find playing with software great fun. This article explores some of its uses, yet an important challenge for the future will be to ponder the implications between those who endorse software wholeheartedly and those who refuse to use it altogether. Will the divergent paths lead to altogether different ways of doing things?

2 For an idea about what these assorted windows look like, it is best to visit the Rearchware website at http://www.researchware.com.

3 Source files must be formatted appropriately for HyperRESEARCH. The software manual is clear on those rules (ResearchWare, Inc., 1999). For example, text files must be in 'text only' or ASCII format. They should be locked. Once you start coding you may NOT alter a source document. Tildes (~) may be added to signify page breaks in the original document. Image files must be JPEG or GIF (or PICT for Macintosh users). Audio or video files must be .AVI or .MOV and HyperRESEARCH uses Quicktime.

4 You would have to code for individual children on each session case card and then use those individual child codes in analysis.

References

Drisko, J. (1997) 'Using Qualitative Data Analysis Software: Merits and Hazards', *Computers in Human Services* 15(1): 1–19.

Liebow, E. (1995) *Tell Them Who I Am: The Lives of Homeless Women*. New York: Penguin.

Miles, M. B. and Huberman, A. M. (1994) *Qualitative Data Analysis: An Expanded Sourcebook*, 2nd edn. Thousand Oaks, CA: Sage.

Researchware, Inc. (1999) *HyperRESEARCH 2.0: User's manual*. Thousand Oaks, CA: Sage.

Schwandt, T. A. (1997) *Qualitative Inquiry: A Dictionary of Terms*. Thousand Oaks, CA: Sage.

Staller, K. M. (2001) 'Doctoral Education at CUSSW: Looking Past the Present', in R. Feldman and S. B. Kamerman (eds) *The Columbia University School of Social Work: A Centennial Celebration*, pp. 347–65. New York: Columbia University Press.

Staller, K. M. (2002) Working the Scam: Policing Urban Street Youth, *Qualitative Inquiry* 8(5): 550–74.

Weitzman, E. A. (2000) 'Software and Qualitative Research', in N. K. Denzin and Y. S. Lincoln, *Handbook of Qualitative Research*, pp. 803–20. Thousand Oaks, CA: Sage.

Karen M. Staller is an assistant professor at the University of Michigan School of Social Work. She holds a JD from the Cornell Law School and PhD in social work from Columbia University. She teaches in the areas of history and philosophy of social welfare, social welfare policy, child welfare policy, and qualitative research methods. She has worked extensively with runaway and homeless youth. Her research has examined the social construction of public problems,

services for at-risk street youth in historical perspective, and the intersection of law and social work. Address: University of Michigan, School of Social Work, 1080 South University, Rm. 2702, Ann Arbor, MI 48109.
[email: kstaller@umich.edu]