

Relational coding: Enhancing the transparency and trustworthiness of grounded theory research

Carmel Rawhani 

Abstract

Grounded theory practitioners are undoubtedly familiar with critiques of their research which problematise its lack of transparency and, subsequently, trustworthiness. While the literature on grounded theory already benefits from numerous varying guides on ‘how to do grounded theory’, comparatively less instruction exists around how to do it clearly and convincingly. One of the elements of the grounded theory methodology (GT) which appears to be impacted most by confusion around transparent writeup is coding, naturally leading to the theoretical outputs thereof suffering from questionable trustworthiness. Through the course of my own postgraduate research, I developed a tool which not only makes coding more transparent to the reader, but to the researcher themselves. This tool, Relational Coding, highlights the relationships between codes which form the foundation of theory, and reveals them more obviously to the reader and researcher alike. Rather than ‘a how to guide to GT’, this is a ‘how to show you have actually done GT’ tool. This allows for stronger arguments to be built from the theory which this methodology builds due to the trustworthiness inspired by enhanced transparency. Here I demonstrate its usefulness and applicability, as well as key shortcomings which may be addressed by fellow grounded theory enthusiasts as the tool is inherently organic and modifiable.

Keywords

Grounded theory, methodology, coding, relational coding, transparency, trustworthiness

Introduction

“more and more, qualitative researchers are exhorted to clearly communicate the logical steps in their data analysis, yet conventions for doing so have yet to appear” (Deterding and Waters, 2021: 710).

While an already well-established methodological framework, Grounded Theory (GT) faces a challenge such that its results are often perceived to have limited trustworthiness (Aldiabat and Le Navenec, 2018; Connelly, 2016; Glaser, 2019; Kolb, 2012; Sikolia et al., 2013; Timonen et al., 2018; Williams and Morrow, 2009). However, a methodology’s transparency and implied trustworthiness are of paramount importance (Aguinis and Solarino, 2019; Jacobs et al., 2019; Maher et al., 2018; Miguel et al., 2014; Moravcsik, 2014, 2020; Tuval-Mashiach, 2017). In light of this established weakness of GT, here I propose Relational Coding as a tool enhancing the methodology’s transparency and trustworthiness by making GT coding more accessible to the researcher

and reader alike. While not a new approach or an amendment, it is a potentially malleable toolkit, rooted in Glaser’s approach to GT coding, which can better display the legitimacy of current dominant approaches by providing ‘an audit trail. . . as a means of holding up to scrutiny the methodological and theoretical decisions made throughout the research process’ (Bowen, 2009: 305).

In aiming for trustworthiness, my understanding draws on Connelly’s (2016) explanation, suggesting that a trustworthy study is one which generates trust in its data, interpretations thereof, and the interpretation process itself. Per Connelly

Wits-TUB-UNILAG Urban Lab, Visiting Fellow, School of Architecture and Planning, University of the Witwatersrand; South Africa Finance and Violence Prevention Portfolio Coordinator, KfW Development Bank, Gauteng, South Africa

Corresponding author:

Carmel Rawhani, Wits-TUB-UNILAG Urban Lab, Unit 308 Killarney Park, 15 5th Street, Killarney, Johannesburg, Gauteng 2193, South Africa.
 Emails: meymoon19@gmail.com; carmel.rawhani@kfw.de



Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons

Attribution-NonCommercial 4.0 License (<https://creativecommons.org/licenses/by-nc/4.0/>) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

(2016), I believe a useful theory-building methodology: produces research that generates comfortable confidence in its findings (demonstrating how they were arrived at); includes an easy-to-follow audit trail which renders the published results dependable; forces the researcher to document subjective decisions and gives the reader insight into these; and richly documents contexts so that the extent of generalisability claims are clear. Considering this, Relational Coding provides transparent foundations essential for theory-producing work like GT where the aim is not necessarily the accuracy of outcomes and their verifiability or replicability based on said accuracy. Rather, GT aspires to demonstrate convincing and explanatory concept-building which becomes increasingly abstract and ends with claims of said abstractions' generalisability (Bryant, 2017; Glaser, 2002b; Maher et al., 2018).

Classical grounded theory is a theory-generating/locating methodology which relies on the disciplined use of constant comparative analysis to uncover concepts from data and locate generalisable theory within the relationships between these data-grounded concepts (Glaser and Holton, 2007). Half a century has passed since Barney Glaser and Anselm Strauss first presented *The Discovery of Grounded Theory* (1967), culminating in its status today as one of the most regularly relied-upon methodologies for qualitative studies. However, confusion arises from the fact that 'grounded theories are neither stories, lists of themes, nor descriptions, rather they consist of parsimonious rigorously extracted concepts bound by inductively derived theoretical relationships', and therein lies a potential cause of misinterpretation as that 'network of concepts and theoretical relationships may be invisible to one who is unfamiliar with the method' (Nathaniel, 2020: 1). The problem, in short, is not that researchers do not know how to do GT, but rather that they do not always succeed in recording, reporting, and therefore demonstrating how they have done it *well* (Charmaz and Thornberg, 2021; Kelle, 2007; Nathaniel, 2021). Thus it is essential to expose 'for both researcher and reader – how the researcher(s) arrive(s) at certain positions during the knowledge construction process' (Subramani, 2019: 1). Importantly, though I use the term 'methodology' from time to time in reference to GT, I am referring to the fact that GT is a grounded theory which informs a research methodology, and it is the methodological aspect of this theory and its weak trustworthiness which I am addressing here.

In the course of my own attempt at learning how to do GT coding, I found multiple inputs rooted in the methodology's various strands from established and emerging voices in the field alike, including: Glaser and Holton (2005, 2007), Holton (2007, 2010), Corbin and Strauss (1990, 2008), Strauss and Corbin (1994), Charmaz (2008, 2014, 2017), Clarke (2019, 2003, 2007), Allan (2003), and Bowen (2009) among others. These various strands are characterised by varying degrees of clarity when it comes to how to code and how to demonstrate that one has done so reliably, leading

many to believe that GT researchers suffer from 'methodological incompetence or fragility' (Bryant, 2002: 32; Glaser and Holton, 2005; Strauss and Corbin, 1994).

The question remains: what can Relational Coding add to GT as it is currently practiced? As a tool in early development it holds the potential for improvement and modification to widen the range of its applicability. Nonetheless, it makes two small but distinct contributions. First, it lays bare analytic functions which do not automatically occur regardless of whether one is coding manually or digitally. Second, a review of recent impactful GT publications indicates that Relational Coding is largely absent from current reporting and writeup practice, and even where scholars hint that they have taken codes' relationships into consideration, there is no demonstration of how this happened. For the purposes of this review I examined articles on the Web of Science database containing 'grounded theory' in the title, restricted my search to articles published in the years 2018–2022 (1236 articles in total), and reviewed the top 10. I then searched through articles published in the last 5 years specifically demonstrating or discussing the outcomes of coding with 'grounded theory' and 'cod*' in the topic (2119 articles in total) and analysed the top 10. I read each article with an interest in its approach to coding specifically, rather than its contribution to grounded theory in general. I examined whether research is laying out blueprints for coding in a more transparent and trustworthy manner, and found that while there are some significant contributions in this regard, the relationship between codes is not sufficiently addressed, per the results captured in Tables A and B in the Appendix.

Positioned within the knowledge gaps outline here, this paper begins with a brief explanation of the importance of coding as my chosen focus within GT literature. I move on to explore GT's original point of departure, the ensuing rift between its founders and the impact that had on coding approaches, and the more recent coding frameworks which have followed. Far from an exhaustive review, this section touches on a few of GT's major strands which have influenced my own understanding insofar as their elements which are relevant to coding in particular. I then go on to briefly explain my own understanding of coding, rooted in Glaser's work, which was moulded under pressure to compromise in some areas where my examiners and faculty required it, during the course of PhD and Master's research. I conclude by drawing on examples from my own research explaining how to write up Relational Coding, with a particular background in the expectations levelled against researchers in academia. I demonstrate diagrammatically how Relational Coding works as a tool for carrying out and writing up the coding process with an emphasis on uncovering the underlying relationships which contribute to theory. It additionally complements existing coding software by proving that coding was supplemented by analysis, and that this analysis is traceable through records, which software alone cannot account for yet (Deterding and Waters, 2021; Maher et al., 2018). It is a

Glaserian-inspired approach to coding which demands a demonstration of the researcher's accountability, rendering the process and writeup of coding clearer to the GT reader and author alike, thereby contributing to the enhancement of GT in general.

Key contributions to GT coding

Importantly, accomplished 'how-to-do-GT' guides already exist, and while touching on some elements of the methodology related to coding I will avoid this task which many scholars have already ably accomplished (Birks and Mills, 2010; Birks et al., 2019; Chun Tie et al., 2019; Flick, 2018; Glaser, 1998, 2007; Sbaraini et al., 2011). Here, I explore the work of a handful of those scholars whose contributions have influenced my own understanding. I review these briefly to demonstrate two points. First, I seek to highlight why coding for relationships is relevant. Second, and building from this, I set out to demonstrate how relational coding functions as a 'how to show you did GT' guide in a manner that is complementary to the various established approaches which I sample here.

Glaser, Strauss, and the discovery of GT¹

In contrast to deduction-heavy approaches typical of academia at that time, Grounded theory was first crafted by Glaser and Strauss in an effort to criticise dominant modes of scientific research. Thus, Glaser and Strauss became active proponents of immersive inductivism as a tool for extracting theory from data. Seeking to depart from research as verification, they pivoted towards theorising as their focus instead and paved the way for questioning the assumed exclusive validity of quantitative research (Chun Tie et al., 2019; Glaser and Strauss, 1967; Timonen et al., 2018).

Grounded theory's core process is abstractive conceptualisation, aiming to produce theories based on the relationships between the most important emerging concepts through repeated rounds of data collection and analysis (Glaser, 2016, 2019). However, despite the numerous detailed components which GT involves (summarised in Appendix B), and its creators' desire to contribute towards frameworks which ensured rigour in qualitative research, 'the methodological advice pertaining to this foundational variant of GT is relatively lightly documented' (Timonen et al., 2018: 2). It is perhaps as a result of this lack of clarity that Glaser and Strauss later learned they differed pointedly on numerous points related to the underpinnings of GT (Howard-Payne, 2016). As such, GTs processes and their implications for codes are discussed separately under the following sections which explore Glaser and Strauss respectively.

The implication for coding as GT's core process specifically is that while its importance is well-established (Glaser and Holton, 2007; Holton, 2007) its 'how to', in terms of carrying it out and writing it up clearly, appears to be less understood (Allen, 2014; Bowen, 2009; Glaser and Holton,

2005). Consequently, GT studies are often recorded in a manner which obscures the research process, suggesting a lack of transparency to readers, reviewers, and potentially even the researcher themselves (Dunne and Üstündağ, 2020; Glaser, 2012; Yu and Smith, 2021).

Strauss and Corbin

It is Strauss, having successfully partnered with Corbin following his split with Glaser, who provides us with early attempts to make coding more accessible in his seminal *Qualitative Analysis for Social Scientists* (Strauss, 1987). For him (1990) GT coding consists of three key elements: open, axial, and selective. These are not necessarily separate phases, but rather various qualities of coding which may be simultaneous and intertwined, or separate. Importantly, for Strauss codes are generated through immersion in the literature very early in research (Strauss and Corbin, 1998, 2008).

First, during open coding, the type of analysis carried out in open coding is purely generative, including questions such as: What is this concept? How does it manifest? What sets it apart from other similar concepts? Axial coding follows from open coding, drawing categories and subcategories into relationships with one another, and testing this against the data itself for verification. The researcher who codes axially must compare the conditions, context, strategies, and consequences relevant to each category and subcategory, to draw them into relation with one another. Without this close reflexive relationship between the data and analysis, the product of GT cannot be anything other than the suggestion of theory with some major gaps. Instead, if axial coding unfolds rigorously, all researcher bias, deductive thinking, and data-based hypotheses are constantly verified- even negative cases are drawn into the concepts to make them richer, and the outcome is powerful explanation (Corbin and Strauss, 1990). Finally, selective coding is the process whereby all the codes are brought together to form a core category.

Corbin and Strauss (2008) provide detailed guides on how to code using software, but fail to capture the iterative induction component inherent to this process. Thus, while mapping out a useful skeleton for GT studies their work lacks an explanation for demonstrating the complex network of connective tissue and muscle which gives the skeleton its form. In an effort to create transparent approaches to writing up codes and theories which are creative, rather than created, I present Relational Coding as one possible tool. Before moving on to demonstrate its qualities, it is useful to consider alternate approaches to coding which view codes as emergent, in addition to smaller contributions to coding from other scholars.

Glaser and Holton

Glaser's approach to GT, which according to him is most true to form and thereby labelled classic (Glaser, 2014), is partially an elaboration and specification of the work he

began to publish with Strauss, and partially built from passionate responses to other strands of GT which he problematizes (Glaser, 2002a, 2019). A number of Glaser's influential contributions have been made alongside Holton (a prolific contributor to GT in her own right) and so I discuss their work together here.

Coding is so important to GT that Glaser and Holton (2007) advocate for it to be a daily pursuit in early research. For Glaser (2019), the key motivation behind coding is the crucial step of moving beyond accurate description, towards theory with explanatory power. Unlike Strauss he advocates against early immersion in the literature as a manner to facilitate coding. At the same time, he does not demonstrate how one would prove they have followed the coding processes he advocates for.

For Glaser, coding has two key qualities/parts (open and selective) and two key types (substantive and theoretical). Open coding identifies core categories in the data which explain exactly what is happening, at a granular level, that is, its substantive empirical qualities. When the researcher is confident that they have located the study's core variable, selective coding can begin, as that is the point at which the researcher will be able to discern which codes relate to one another in a manner which produces theory. Through selective coding theoretical codes begin to emerge. Their effectiveness and usefulness rests on their conceptual and abstract nature (Glaser and Holton, 2005). Theoretical coding foregrounds the question of relationships, particularly those between the substantive codes discovered earlier. These relationships, fully conceptualised during theoretical coding, are best understood as early hypotheses which come together to become the building blocks of theory (Glaser, 1978). Notably, Glaser and Holton emphasise that theoretical coding is perhaps understood most weakly, and is most often captured obscurely in a manner which doesn't always demonstrate what the codes are *grounded* in Glaser and Holton (2005).

The entire process rests on *selectively* pulling the codes into groups for comparison and showing that this has been done as a foundation for demonstrating that the subsequent emerging ideas are all truly grounded in the data from which they emerged (Glaser and Holton, 2007). The final theoretical code is one phenomenon which relates all of the substantive codes to one another (Glaser and Holton, 2005). Glaser's (1992, 2002a) strong emphasis on avoiding forcing any ideas or relationships is why writing up this process clearly is key (Glaser and Holton, 2005).

In terms of write-up, Glaser and Holton (2007) highlight the need for coding to be captured in a manner which demonstrates the connections between all the component concepts of theory. The emergence and introduction of an idea should be traceable throughout the various levels of theory (Glaser and Holton, 2007). However, a clear step-by-step example of how to achieve this is lacking. In this context the clear demonstration not only of these relationships but also of their grounding in the codes and the data is what Relational Coding focuses on, hoping to contribute to this gap.

Charmaz, Clarke, Allan, and Bowen

Without wanting to reduce the grounded theory academic discourse to the works of Glaser, Strauss, Holton, and Corbin, briefly touching on a few additional scholarly contributions to GT coding is useful without deeply exploring the work of each author. Kathy Charmaz, who has also worked with Linda Belgrave, Antony Bryant and Robert Thornberg (Bryant and Charmaz, 2007, 2019; Charmaz and Belgrave, 2019; Charmaz and Thornberg, 2021), emerged as a leading proponent of constructivist GT. She proposes coding in line with the logic of constructivism which frames the researcher as an active shaper of the data rather than a removed observer (which has controversially drawn the ire of Glaser (2002a) while being hailed by others as most useful (Birks et al., 2019)) (Charmaz, 1996, 2011, 2017).

For Charmaz and Thornberg (2021) early coding is very intense as it involves a line-by-line approach of slow, focused, and deliberate engagement with the data at a granular level, an essential step to confident coding later on. This line-by-line approach can stop when the researcher discerns how codes come together and which ones 'feel' the most important. Sorting the codes is key to deciding on their importance, because codes (per Charmaz and Thornberg, 2021) are not all alike. Those codes with the ability to explain more of the data than other codes can be prioritised as 'important' or 'focused' codes, whose explanatory power must be tested against larger and larger portions of the data. This process is called focused coding, and when followed properly it can speed up analysis by generating possible categories for further exploration early on. This directs future coding to focus on the questions and themes generated during focused coding. Based on Charmaz's approach to coding Ylona Chun Tie, Melanie Birks, and Karen Francis (2019) have created useful graphical representations of the various phases of coding. However, these adaptations only serve to demonstrate to the researcher the relationship between various steps of coding, rather than showing the researcher how to demonstrate to the reader the relationship between the various *codes*. This highlights a useful space for Relational Coding's contributions.

As opposed to the explanation-centric focus of coding per Glaser, Strauss, and those who have worked with them since the publication of *Discovery* (Glaser and Strauss, 1967) Charmaz's constructivist approach to coding focuses on producing theory with an interpretive and abstractive quality more so than explanation (Charmaz and Thornberg, 2021). Charmaz and Thornberg (2021) suggest that shedding light on the author's positionality and epistemology are of key importance to the study, and in this way they provide a proposal for enhancing the transparency of the coding process indirectly. While this task would go beyond the scope of this paper, models which could demonstrate how to transparently and effectively work one's research philosophy into the GT framework insofar as coding decisions especially, are of paramount importance. Finally, Charmaz raises the call for

enhanced transparency in GT, stating that ‘the transparency of the method enables researchers to make transparent analytic choices and constructions. The researcher can see and create a direct relationship between data and abstract categories’ (Charmaz, 2008), however this stops short of indicating exactly how this transparency could unfold. It is this exciting task, building on the organic possibility for transparency inherent to GT, which Relational Coding takes up.

Another influential iteration of GT is found in Clarke’s situational analysis as grounded theory, an approach inspired by postmodernism (Clarke, 2003). Not to be viewed as a standalone take on GT, Clarke’s situational analysis is a supplementary tool to Strauss’s existing work (Clarke, 2003). Starting with open coding, Clarke advocates for an approach which begins word by word and then evolves to looking at the data segment by segment. For Clarke, these open codes are temporary labels whose individual properties are slowly developed and explained as the researcher delimits their scope and begins to decide which codes are important and which are not.

GT researchers, per Clarke, look to find simple yet pivotal social processes underlying the area of investigation, capturing these as gerunds which indicate action. In turn, each process is then understood and described in terms of its ‘particular and distinctive conditions, strategies, actions, and practices engaged in by human and nonhuman actors involved with/in the process and their consequences’ (Clarke, 2003: 558). Codes which are deemed important are related to one another by the researcher, and then combined to make categories which provide a theoretical perspective on the area under investigation. Eventually, this builds towards substantive theory (Clarke, 2003).

Working to enhance what she describes as GT’s existent postmodern potential, Clarke seeks to go beyond the end goal of following the method to produce grounded theory, rather seeking to engage in grounded theorising throughout the process. In order to analyse the data in a fresh way and open it up, Clarke proposes situational analysis via mapping as a key tool. However, per her own admission, this is a tool for processing traditionally coded or at least partially coded data. In the same breath she notes that GT needs a postmodern orientation so that it can acknowledge the complexity, differences, and necessary relationality of postmodern reality. Without exploring Clarke’s mapping tools, which are certainly valuable additions to the GT discourse, they are not an approach to coding. However, her work sparks important questions about the methodology while leaving a gap where coding which appreciates the inherent relationality of the subject of analysis could make a significant contribution. While making great strides in enabling the researcher to understand the delimitations of coding, Clarke’s method does not indicate how to demonstrate to the reader that this delimitation is rooted in queues from the data itself. Here, I present the potentialities of Relational Coding.

Finally, a number of contributions have been made to GT through its application to information systems and computing. One useful attempt to clarify coding is rooted in the work of

Allan (2003) who utilises Key Point Coding. For Allan (2003) coding became too cumbersome at the micro level, leading him to focus on coding on a key-point-by-key-point basis. This approach begins with identifying important incidents to code and assigning them an identifier which allows the researcher as well as the reader to track the origin of the code. Each identifier consists of two parts: first, a sequential identifier indicating when it emerged (first point=P1, second point=P2. . .etc.). Second, longitudinal identifiers are added to indicate the case study/site in which the point emerged (so key point 5 emerging in case study Z would be written as P_Z5. Associated codes/explanations would then be written alongside each identifier, so that each code can be traced to its point of origin. Building on this first phase of identifiers and codes, codes are then compared to one another (a process which can be tabulated quite neatly using all of the identifiers) so as to locate commonalities and produce concepts, before building towards theory. Allan’s (2003) use of diagrams and tables provides useful insight into coding and Key Point Coding in particular. While it demonstrates a great awareness of constant comparative analysis and the process of drawing codes and concepts into relation with one another, it appears to stop short of indicating how to render these relationships transparent. This approach shows how codes are grounded in data, but not how relationships are grounded. It is those relationships which are a basis for theory, confirming the useful potential contribution of Relational Coding.

Also seeking to demystify coding, Bowen’s (2009) proposed audit trail technique explicitly lays out a tool to systematically trace research as it processes data, from early collection to final analyses. Bowen insightfully highlights a key hurdle for GT studies: replicability. His most powerful contribution is in demonstrating a tool which allows for replicability of process, without expecting replication of results (Bowen, 2009). In this manner, the steps of the researcher can be easily traced, as with Allan’s (2003) Key Point coding technique. Going beyond a mere transcript of the research, the idea of an audit trail is rooted in the need for GT to render the analysis process transparent and trackable. For Bowen, this can be demonstrated diagrammatically showing which data each code comes from, which open codes become which axial codes, and which axial codes in turn create a selective code (Bowen, 2009: 314). What this stops short of, however, is elaborating on *how* the codes become condensed and raised to higher levels of abstraction, and without this relational aspect a key element of transparency remains missing. While Bowen’s own theory-building work doesn’t leave out relationships per se, the audit trail tool omits a clear method for recording and explaining the relationships, rather than just pointing towards them, and Relational Coding seeks to make a contribution in that space.

What next?

The preceding sections established the importance of research transparency and trustworthiness, the strengths and

limitations of current approaches to GT and their implications for coding, the weakness of GT coding whose writeup and execution are vulnerable to limited transparency and trustworthiness, and a lack of existing GT papers devoted to addressing this. With regard to Strauss and Corbin's work, I find myself agreeing with Charmaz and Thornberg's (2021: 7) assessment of it as 'rigid and prescriptive in sharp contrast to the open-ended, fluid approach' of earlier publications. In reflecting on Glaser and Strauss's (1967) early work, carried forward in some senses by Glaser (1992, 2014), Glaser and Holton (2004), I find the element of inductivism over-exaggerated, or rather under-specified (cite). Overall, the approaches reviewed here rely on delimitation as a strategic research tool, without explaining how exactly to demonstrate such delimiting decisions on the part of the researcher. Inspired by Bowen (2009) and Allan's (2003) respective efforts to create an audit trail for GT, I share Relational Coding as GT demonstration toolkit designed to create such a trail for the researcher and reader alike by contributing to the gap in knowledge and practice where code relationships are concerned.

In presenting this toolkit I acknowledge that Relational Coding is not a one-size-fits-all fix for GT or methodological transparency in general. Per Corbin and Strauss GT is *'reproducible in the limited sense that it is verifiable. One can take the propositions . . . and test them. However, no theory that deals with social psychological phenomena is actually reproducible in the sense that new situations can be found whose conditions exactly match those of the original study'* (1990: 15). This strand has held true throughout Corbin and Strauss's work, noting that the idea behind coding is that *'any conceptual label reflects my interpretation of what is being said, as other researchers may have their own ideas and even disagree with me. The idea here is not to quarrel with everything that I say, but to note the process that I am going through'* (2008: 162). This speaks to Glaser and Strauss's original ideals for GT, as in Discovery (1967) they advocate for grounded theory to be judged on *how* theory emerges. When coding is carried out and/or written up in a manner which obscures this important work, that becomes impossible. As such, I do not pursue transparency and trustworthiness for the purpose of producing reproducible results necessarily. Rather, per Bowen (2009), I aim to clarify the process of coding for the purpose of methodological writeup which enables studies to be emulated. Thus, I propose Relational Coding as a tool for enhancing transparency and trustworthiness in GT in this manner, as one of many possible solutions to this methodological imperative.

Relational coding

Why relationships?

The label 'Relational Coding' may seem unnecessary for a theory-generating process given that theory can be described, simply, as the range of possible relationships

between concepts (Strauss and Corbin, 1994). The problem, as established, is the questionable transparency and trustworthiness of many studies which set out to follow GT. I posit that this is because existing approaches do not adequately indicate how to demonstrate the use of relationships as foundational for coding and theory, much less explain why those relationships are plausible and explanatory based on the data they supposedly emerged from. While I believe that many of the existing approaches to coding *do* allow researchers to achieve this, I propose Relational Coding as another contribution which seeks to facilitate this task on the part of researchers who resonate with its principles.

Inspired by Corbin and Strauss's (1990) assertion that 'if a grounded theory researcher provides the pertinent information, they enable readers to assess the adequacy of a complex coding procedure' (p. 17), my approach to Relational Coding is rooted not only in a desire to facilitate rigorous, coherent coding, but to enable myself (and all who may choose to utilise this) to write up the often mystical coding process with as much useful detail as possible, without burdening the reader with a bulk of data which obscures the process of analysis. As such I set out to code in a manner which weaves in constant reminders to theorise into the process and have captured this within Relational Coding.

Per Appendix B's Figure A, my personal take on GT is closely aligned with that of Glaser, though I also benefit from the contributions of his peers as mentioned previously. I use open, intermediate, and theoretical coding (alongside a late-stage literature review) to locate emergent theory. Rooted in Glaser and Strauss's (1967) early work I find it most useful to code at the level of incident, rather than word by word or line by line. The following section elaborates on Relational Coding in full, before moving on to share examples from my own previous research which serve to demonstrate what it might look like.

What does Relational Coding look like?

At the outset it is important to specify that Relational Coding is applied across all of the, per Glaser, 'classical' phases of GT coding (Glaser, 2014). In accordance with Glaser's instruction to keep coding open in the beginning, the beginning of Relational Coding are loose and free though structured by an internal logic (my personal preference is abductive logic (Reichert, 2007, 2019)). For open coding this practically means identifying themes as regularly or irregularly throughout the text as makes sense and recording them all in the form of memos which contain explanations for each theme that allow you to return to the text where the code originated. I analyse all data, when I start coding, with my problem question in mind (one which changed in and of itself throughout the research process). When I choose incidents to code which are relevant to what I am very generally looking to learn about, I record my codes and memos in

answer to the question: what about this interests me and why? (this question might change from one author to another, but clearly acknowledging it is key to providing methodological transparency). I choose to do this by inserting text into pdf versions of the documents I analyse, as comments in the margins, which were easily searchable, or as voice recordings and written notes stored in my cell phone during immersive observational fieldwork.

When I have coded in this manner for a few months (or as long as is needed to reach a point where coding is generating bulk rather than new substance) and coded for a few successive rounds (searching for data based on where the previous data suggested I should look for more answers) I pause the process. Here I review all the codes, following which I gather them into one document. Usually at this point I discover that I have hundreds of proto-codes (early versions of codes, or prototypes). While GT encourages free thinking it also requires avoiding bulk in the data (Barlett and Payne, 2004), and so at this point, keeping in mind the numerous questions that Glaser (1992), Corbin and Strauss (1990), and others suggest posing to the data, I find two to offer a particularly useful foundation for reducing code bulk: first, what is happening here (referring to process), and second, to what end/meaning (referring to produced meaning)? Far from restrictive or exhaustive, this short list of questions, for me, encompasses a much longer list including: What is the main idea being studied? How could I summarise it? What do all of the actions and interactions I studied lead back to? How can I explain outliers? (Corbin and Strauss, 1990). What is this data studying? What is the emergent category? What is happening? What is the main problem/concern facing the people I am studying? And how is this resolved/attempted to be resolved? (Glaser, 1992).

This set of questions lies at the heart of Relational Coding (though it may be modified to include different or additional questions as per any study's needs). In no way is this intended to limit the scope of discovery and, consequently, theorising. However, as with every research design choice a researcher makes, these questions inevitably have a delimiting effect on any study and thus must be listed clearly and referred to transparently. In the case of my own previous work I have always listed the questions in full and opted to attach appendices summarising which questions dominated particular stages of coding to provide further insight into the subjective and iterative processes behind coding. Ultimately the reduction of the rich range of coding-guiding questions emerging from the works of Glaser and others is a choice which aims to make research and its writeup easier while avoiding limiting its depth. Asking 'what is happening' and 'to what end/meaning' offers a way to summarise, from my point of view, the full range of questions which have been drawn on. These other questions are not erased. Instead, reflections around those questions are captured in memos and recorded in the writeup whereby the definition of each code is provided. For example, questions such as 'what is the problem here?' and

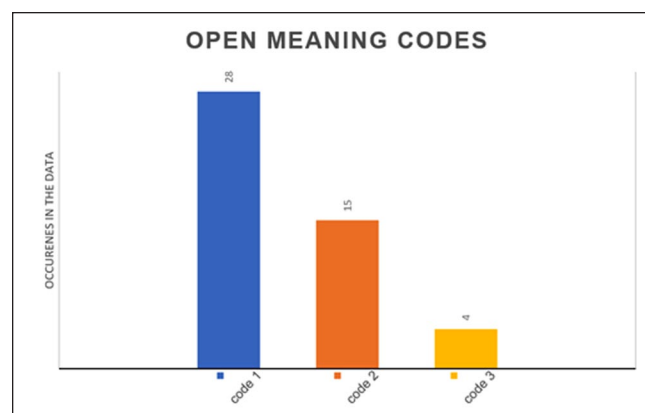


Figure 1. How to display Open Meaning Codes displayed by frequency of occurrence in the data.

'how is it being solved/ignored?' are implied corollaries of what is happening. At the same time, as mentioned in the paper, I envisage Relational Coding as a malleable framework which can respond organically to different priorities, interests, and capacities of different researchers. As such, lists of questions and the manner in which they are recorded will differ from researcher to researcher, but tying the coding to the questions asked is an essential step in ensuring transparency and trustworthiness.

In the case of my own work with Relational Coding, I pose these two questions (what is happening, and to what end/meaning?) to every single one of the hundreds of codes I locate during early coding. The result is often hundreds of lines of two codes each which allow me to explain, at every interesting moment in the data I analysed, what is going on and what is the impact generated. Most recently this significantly reduced the number of codes I was dealing with from hundreds to less than 40. At this point, I re-code the data so that in all of those (potentially) hundreds of incidents two of my new codes could be used as a label (one answering 'what is happening' the other answering 'to what end'). I then continue coding until I reach data saturation (when the data persistently fails to yield new insights compared to the existing coded material (Glaser, 2012)). Often by this point there are thousands of lines of code, each consisting of two codes (usually stored in an excel table or any coding software you may prefer).

Now I begin to consider the thousands of relationships represented in the thousands of lines of codes. To record this, I create tables and bar graphs demonstrating the number of times each code appeared; its emergence relative to other codes (which codes coincide, and how frequently or infrequently?); and the correlation of the various codes in connection with their coincidence. Figure 1, Tables 1, and 2 which follow demonstrate this at a very basic level.

How to display Open Meaning Codes displayed by frequency of occurrence in the data. I would begin to shed light on the data, the codes, and the relationships between them

Table 1. How to record the origin of codes for comparison purposes.

Source	Incident	What is happening? (Process codes)	To what end or meaning? (Meaning code)
Policy document x	Brief description	Open code 1	Open code 4
Park y	highlight	Open code 2	Open code 5
Interview z	keywords	Open code 3	Open code 6

Table 2. How to display codes which occur together frequently for comparison purposes.

Open process code 2	
Open meaning code	Coinciding occurrences
Open meaning code 3	19
Open meaning code 1	14
Open meaning code 2	1

with a chart tallying the total emergences of each code, one for process codes and one for meaning codes. For example, the following Figure 1 displays an example of how meaning codes might be displayed as derived from open coding. In the real writeup rather than the labels ‘code 1’ etc., the name of the code would appear.

From a simple and clear table such as Table 1 I might draft a separate table for each code, isolating that code’s emergences from the data, and identifying which codes it occurred alongside, for both process and meaning codes. These representations of data would then accompany detailed descriptions of each code.

Based on representations such as these, the foundation is laid for rich description and definition of each code rooted in my interpretation of its relationship with other codes, both strong and weak. Having demonstrated these relationships in terms of frequency of emergence the process of defining the codes then demonstrates my subjective interpretation of the data. Here I allow the definitions to take shape around the full gamut of questions posed to the data. In this manner theory is not limited and can achieve richness in a transparent manner that allows for complexity while reducing confusion. It must be stated that this is not an argument on the role of quantitative data in qualitative research. Representing the data with relation to frequency of occurrence should by no means be framed as attributing particular importance to themes. Instead, it is part of an exercise in transparency that I have come to carry out as directed by regular requests for such from reviewers.

In representing the data thusly, two core concepts must be emphasised. First, coding is largely subjective and recording any part of the coding process is a record of biased decisions that enable any reader or reviewer to follow the author’s logic, rather than to test whether it is accurate or not. Second, no false importance should be associated with frequency. For example, a code occurring infrequently, for example, might

Table 3. How to display the beginning of intermediate coding.

Open codes	Intermediate proto-codes
Open meaning code 1	Intermediate meaning proto-code 1
Open meaning code 2	
Open meaning code 3	
Open meaning code 4	Intermediate meaning proto-code 2
Open meaning code 5	
Open meaning code 6	

be even more significant than one occurring very frequently. Additionally, the frequency of occurrence is what allows me to move on to explain co-occurrence, that is a relationship, as I perceive it. This is what demonstrates the subjective choices made in trying to relate codes to one another in a manner that explains the larger questions at hand. Only through representing codes by frequency of emergence in addition to writing up detailed definitions can a fuller range of the subjective delimiting decisions made by the researcher be laid bare for the reader.

Having teased out these relationships, what remains is to begin integrating the potentially thousands of lines of coding and numerous codes into fewer and fewer key concepts. This point represents the end of open coding for me, and the beginning of intermediate coding. Here, I begin by summarising the open codes under a reduced number of labels. For example, recently I combined 14 open codes under the umbrella of three interrelated intermediate proto-codes (prototypes of what the final intermediate codes will be, as I am not finished with the process of narrowing down at this point). The process of beginning to code to unite in this manner relies on simple diagrams once again. First, tabulating the process per Table 3:

These codes, however, are not merely being grouped for the sake of reducing their number. Rather, they are being drawn together and united under new conceptual explanations, capable of generating explanations in and of themselves. For this reason, it is essentially to follow up the simplistic summary portrayed in the tables, with a more powerful conceptual explanation of each intermediate code, showing how the open codes which led to its creation interrelate. Multiple data visualisations would be useful here, though I tend towards colour-coded doughnut charts for their simplicity. The following Figure 2 demonstrates how I display the intermediate proto-codes which I might chose to group together in perceived interrelationships.

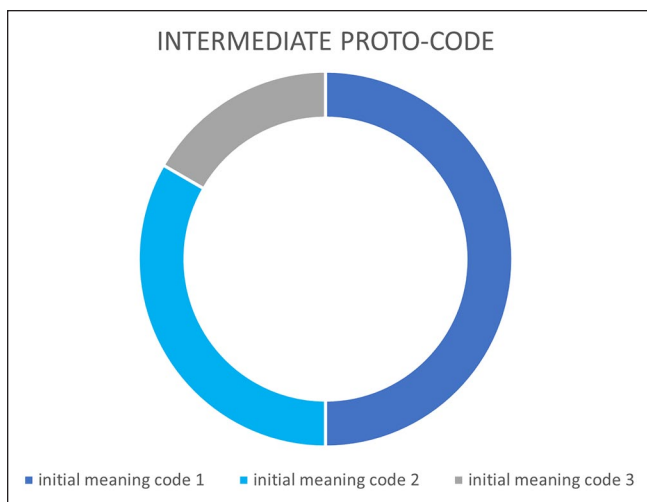


Figure 2. How to display the component elements of early intermediate codes.

Diagrams such as these doughnut charts allow the reader to go one step beyond understanding which open codes are combined to create early intermediate codes. This demonstrates *how* the open codes began to come together in terms of the relative strength of emergence of each concept in the data (the number of times it occurred). This also allows for early definitions of the proto intermediate codes to emerge, both for the meaning and process codes. The next step is to build further complexity in the codes in terms of their richness and their explanatory capacity. In order to do this (See Figure 3) I begin to unite intermediate meaning codes with intermediate process codes, taking care to show how all of the open codes contribute to this meaning-making and explaining, rooted in diagrammatic representations. The following Figure 3 demonstrates a simplified version of how I might writeup this process, ensuring that simple colour-coding is used to make the relationships between codes clear to the reader. Importantly, the code towards which the arrow points is the code whose relationship I am examining, here directed by frequency of co-occurrence, with the code from which the arrow originates (I also do not exclude codes with infrequent co-occurrences as that is also interesting and forms part of the descriptions and explanations that would accompany the writeup of such figures).

Here I use arrows to indicate which meaning codes the process codes are related to most strongly, and vice versa, to justify the explanations that I build from those relationships, and the manner in which I summarise closely related codes into more powerful concepts. Using arrows to demonstrate the relationships between codes which I unite under early concept labels, I seek to lay the foundation for further theory-building that can only happen by continuously narrowing down various concepts into finalised intermediate codes, prior to theoretical coding. In this manner I have previously reduced six early intermediate codes (3 intermediate

meaning codes and 3 intermediate process codes) down to three overall intermediate codes. In doing so their richness and explanatory power increases.

Adding detailed explanations to each set of interrelated codes, I assign a final intermediate code to each combination of intermediate meaning and process proto-codes. This label is derived from the in-depth and transparent explanations I have already provided for the proto-codes, the intermediate codes from which they emerge, and the relationships captured within them. For me, this marks the point at which I am ready to begin coding theoretically.

Finally, in theoretical coding, I narrow down the intermediate codes. Having demonstrated in great detail the emergence of the codes, the relationships between them, and the theoretical building blocks emerging therefrom, I use theoretical coding to map out the narrative generated by these powerful theoretical emergences. Here I seek to locate one overall code with explanatory power and theoretical possibilities, and to take the reader along with me, hand-in-hand, so as to demonstrate my own subjective decision-making process as I pursue one overall theoretical proposition. In this paper I do not explain the narrative work of theoretical coding in great detail as I am not concerned with creating a how-to-code guide, but rather how to code transparently. For me, theoretical coding elevates all of the ideas from previous coding rounds to increasingly generalisable and applicable ideas with enhanced explanatory prowess, most usefully written up as a narrative which refers back to the data from which it emerged. If open and intermediate coding have been carried out as transparently as possible, with every delimiting decision and every subjective choice on code labels, relationships, and the meanings derived therefrom explained in a detailed fashion, then theoretical coding and its results become all the more trustworthy. The entire process is summed up diagrammatically in Appendix B's Figure B.

A practical demonstration

To better introduce Relational Coding I find it helpful to draw from previous studies where I adopted this technique. Without diluting this paper's focus by explaining another entirely, the research in question brought together grey literature analysis and the writeup of immersive observational fieldwork in public parks, exploring inclusion and cohesion in Johannesburg, South Africa. The question which guided the study was: how does the South African government's normative policy vision for post-apartheid public space interrelate public space as it is lived and perceived in Johannesburg? I was specifically interested in four of Johannesburg's public parks representing different neighbourhoods in terms of linguistic, racial, ethnic, and income demographics. The parks themselves were of different sizes, ranging from some being recently entirely rebuilt, and others in various states of decay.

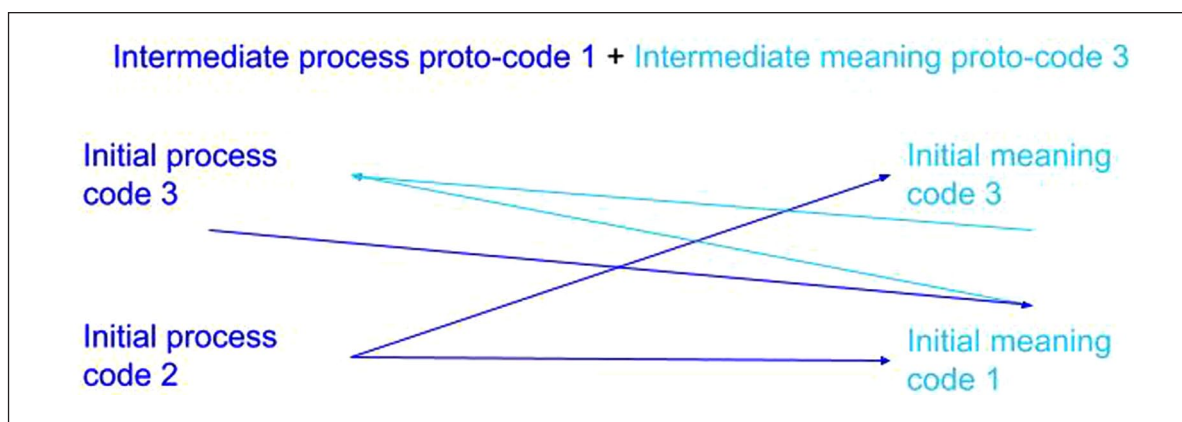


Figure 3. How to display the component elements of early relationships between intermediate codes.

	A	B	C	D	E	F	G	H
1	SOURCE			CODE		EXPLANATION/MEMOS		
2			processual code	meaning code				
3			did what/what's happening	to what end/meaning?				
4			norm creating	space creates community		no space means no community		
5			norm creating	space creates community				
6			problematising	unrealistic policy		unimplimentable policy		
7			problematising	unrealistic policy				
8		conceiving a	conceiving	local as part of national		the local vision fits into/is part of the national one		
9			conceiving	state-set vision				
10			legislating	local as part of national				
11			prioritizing spatial transformation	spatial transformation=urban development				
12			prioritizing spatial transformation	spatial transformation=urban development				
13			prioritizing spatial transformation	spatial transformation=urban development				
14			idealising	space as transformer				
15			idealising	space as transformer				
16			idealising	space as transformer				
17			norm creating	realistic expectations/stifled imagination				
18			norm creating	space as utopia				

Figure 4. A screenshot taken from the master coding document on Microsoft Excel showing codes (sources blurred for anonymity) (anonymised).

Here I began with 2192 lines of code, each containing two codes (one representing process, and the other meaning), made up of 39 individual open codes overall, with each code's frequency of occurrence mapped out per Figures 4–6. Figure 4 specifically provides a screenshot of manual coding and memoing, clarifying how codes are recorded and demonstrating to the reader that a dataset can be found wherein the subjective decisions made by the researcher are accessible. Figures 5 and 6 respectively demonstrate how to potentially display sets of codes which emerged in associated with posing particular questions to the data.

The full details of these codes and their occurrence alongside one another was also made available in tabular form, with an example in Table 4 which follows (adding a

Table 4. Example of mapping out an open process code with the codes occurring alongside it (anonymised).

Open process code: practising (278 occurrences)	
Related open meaning code	Frequency
agency/placemaking/ownership	62
inclusive urban citizenship	44
old norms	42
space as amplifier/multiplier	41
conditional membership of public	32
new norms	25
space as service	22
space creates community	6
space as enabler	4

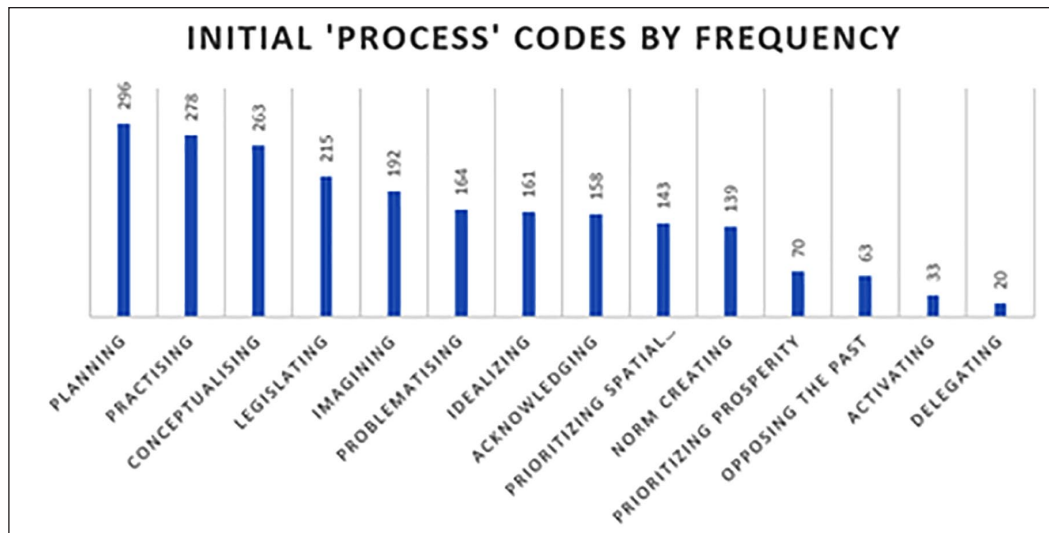


Figure 5. Example of process codes displayed by frequency of occurrence in the data (anonymised).

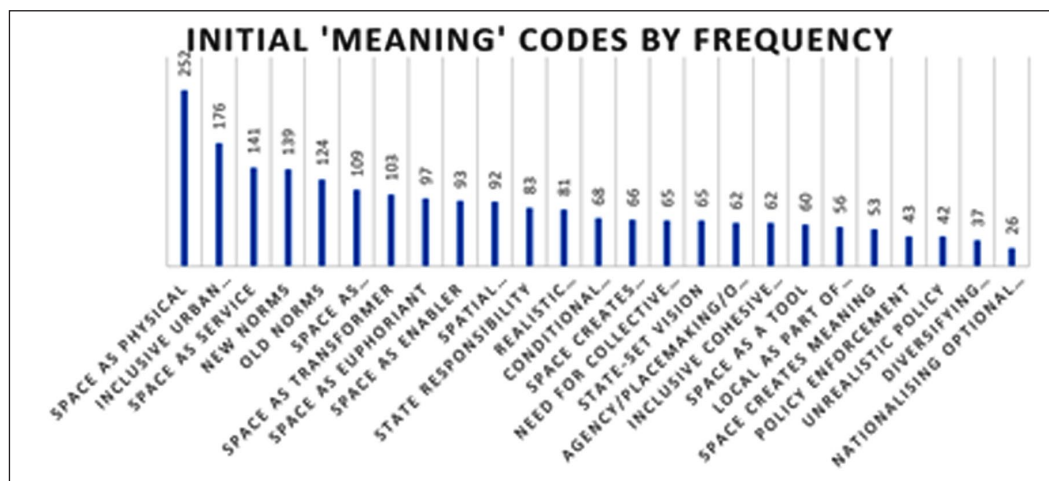


Figure 6. Example of meaning codes displayed by frequency of occurrence in the data (anonymised).

level of complexity to the simple frequency of occurrence of each code on its own as demonstrated already in Figures 5 and 6)

By way of analysis, I identified each code in rich detail, drawing on where and how it emerged from the data, as well as its relationship to other codes which occurred frequently alongside it (per Table 4 above), I was able to narrow down the 39 codes (14 process codes and 25 meaning codes) to six intermediate codes (three process codes and three meaning codes). This was simply done through asking, what are the key processes occurring across the 14 open process codes? The result was three intermediate codes which arose from defining and outlining every single open code (with further explanation provided in the figures which follow from Table 5):

I then focused on each individual intermediate code showing the open codes which combined to reveal it and reflecting on the influence each open code had in relation to the others which together made this singular intermediate code.

Table 5. Example of open codes and the intermediate codes which emerged from them (anonymised).

Open process codes	Intermediate process code
acknowledging	1. envisioning 2. translating 3. creating
activating	
conceptualising	
delegating	
idealising	
Imagining	
legislating	
norm creating	
opposing the past	
planning	
practising	
prioritising prosperity	
prioritising spatial transformation	
problematizing	

This process necessarily included textual descriptions alongside diagrams in order to foreground the relationships between the codes in the diagram (see Figure 7 which follows). For example, Figure 7 which follows demonstrates how an intermediate code, here ‘envisioning’, is comprised of six open codes, which all emerged from the data in various

frequencies and are all significant in different ways. This would follow, of course, from the sections defining each code and explaining how it answers the questions posed to the data.

A figure such as this would be accompanied by explanatory text, as per the following box:

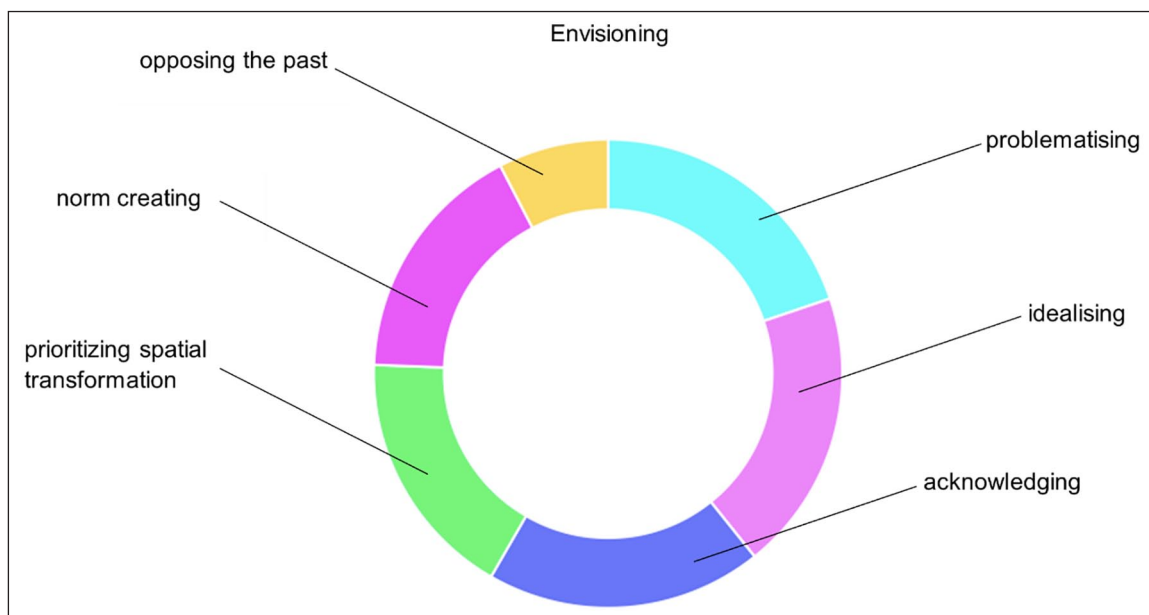


Figure 7. Example of an intermediate code containing the open codes from whose combination (i.e. the relationship between codes) it emerged (anonymised).

Figure 7 indicates the proportional impact each emergent initial code had on the overall understanding of the concept of ‘envisioning’. It is an indication of the fact that, for example, ‘problematising’ occurred far more frequently than ‘opposing the past’, which explains why it is assumed that ‘problematising’ has a greater impact on ‘envisioning’ than ‘opposing the past’. This does not mean that infrequency of occurrence leads to assumptions that codes like ‘opposing the past’ are less significant analytically. As explained previously, the infrequent occurrence of a code can reveal much about it, and in the context of exploring relationships the comparisons drawn from the various degrees of frequency of occurrence are informative as to the various power balances and imbalances between the initial codes.

‘Envisioning’ speaks to an idea which encapsulates the vision for what sociospatial transformation should be. It is both visionary and reactionary in that it idealises and acknowledges new norms and the necessity of their creation as a tool for opposing the past, but it is also reactionary in that it is rooted in the problematisation of space as physical. Inherent within this is a tension between the idealisation and acknowledgement of normative elements of sociospatial urban transformation, and a practical prioritisation of spatial transformation to the exclusion of the ‘social’ element. While problematising the past, very little of its focus is on opposing the past, despite acknowledging that apartheid-era norms endure and that one of the core tenets of the vision is the creation of new norms. This is yet another example of the tension inherent at this level.

Finally, in preparation for theoretical coding, relationships between intermediate codes emerged (based on the relationships between the open codes which made up the intermediate codes) which allowed me to combine the six early intermediate code into three early theoretical codes, resulting in one final overall code. Figure 8 which follows is an indication thereof. As demonstrated in the figure, colour-coding provides an easy manner in which to highlight the perceived relationships between sets of codes, further

demystifying the subjectivity of grounded theory work. Of course, this is only possible if rich explanations have accompanied the mapping of each intermediate code as demonstrated with Figure 7 and in the box following Figure 8:

In a standard writeup a diagram such as Figure 8 might be accompanied by 3–5 pages of analysis and explanation, however for the purposes of the scope of this section within a larger paper with a different focus, a short excerpt is in the following box:

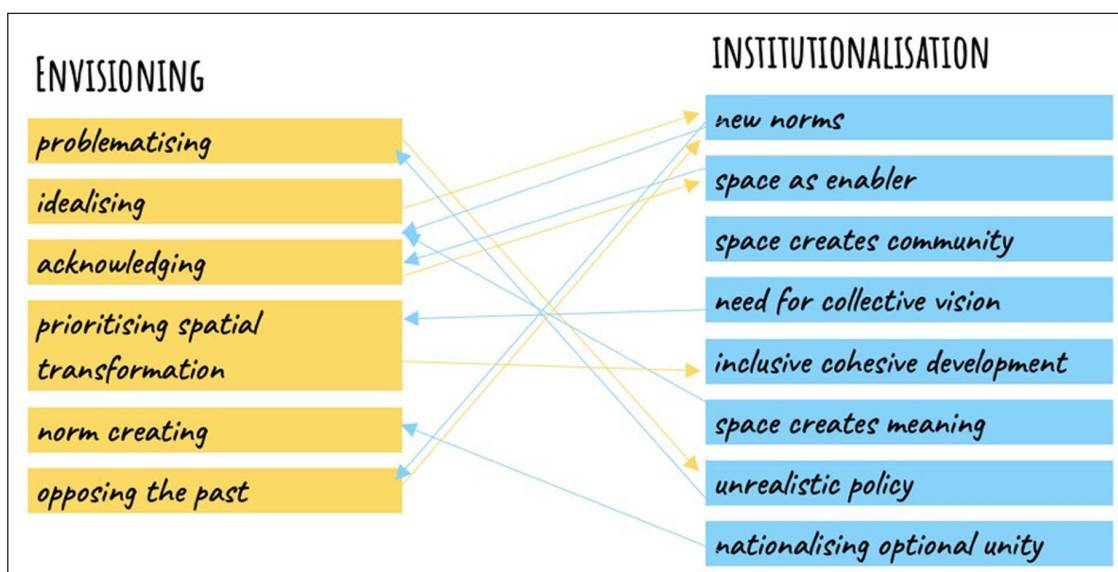


Figure 8. Example of an intermediate code containing the open codes from whose combination (i.e. the relationship between codes) it emerged (anonymised).

The institutionalisation of the vision of cohesion-generating inclusivity in Johannesburg's public parks results in a space > public dichotomy whereby public space is planned in terms of space first and public last. The processes and meanings underlying this are shown in Figure 8. Here, the institutionalisation of envisioning effectively outlines how to pursue normative sociospatial transformation in urban public spaces in South Africa. It is rooted in an idealisation of new norms that leads to a focus on norm creation. These new norms are idealised in a sense that their creation or acknowledgement is deemed to be enough to create change. There is an acknowledgement closely linked to this appreciation for new norms, emerging from the data I analysed, that many of the norms witnessed in South African society today are more reminiscent of apartheid-era beliefs and behaviour. This emphasis on creating new norms, without effectively recognising the endurance of old norms as a problem compromises the validity of the vision and gives it a fatal flaw, because it creates tension between the normative vision desired, and the problem which is located as an obstacle to that vision.

The problem, which is highlighted first and foremost in the data analysed, is that policies are inadequate. While this is not inaccurate, it subverts the aims of normative policy by directing the focus for points of intervention outside of government and outside of policy, to the environment within which policy is enacted, blaming corruption, a lack of accountability, a shortage of resources, limited resources, and the legacy of apartheid and colonialism (without recognising the norms of apartheid and colonialism as problems to be addressed). Because of this, the new norms which are created emphasise inclusion, rather than cohesion. They emphasise the need for a collective vision as a solution to sociospatial problems, thereby promoting the national identity project over social solidarity and unity at the local urban level. Furthermore, this idea of the need for a collective vision is out of touch with policy architecture, because trusting a collective vision to effect public changes implies that the collective who own that vision will have the ownership to shape it and use it in an actively participative manner. Indeed, the overwhelming alienation of individuals and groups outside of the state from setting this vision has meant that they accept the vision of the state, whatever it is, and do not view it as their responsibility to use their agency to create change as it is not even acknowledged.

The intermediate codes 'envisioning' and 'institutionalisation' were, in this case, part of six early intermediate codes (or codes that emerged early in the intermediate coding process) that were narrowed down to three final intermediate codes. Per Figure 8, each final intermediate code emerged from the analytic pairing of a process intermediate code with a meaning intermediate code. In the case of 'envisioning' and 'institutionalisation', for example, the emergent intermediate code which I found best captured what I had mapped out, including both relationships and the meanings behind them, was the idea of 'a space > public dichotomy'. In other words, the idea that the institutionalise vision generated is one which prioritises considerations of space above considerations of the public in public space.

Building on the relationships visually mapped out in this way, the path is paved for engaging in rich descriptive and explanatory code discussions as one coherent theory which cohesively bring all of the codes together emerges. In this particular study I was able to discover that public space policy lacks considerations of the public as their agency is hidden and assumed to be superseded by the 'space' that is public space (anonymised). Using this approach, the combinations of relationships between codes can be laid clear, so that the emergence of theory is not shrouded in mystery, and the choices of the researcher underpinning each step of analysis are laid bare. The following section builds on this high level summary by instructively mapping out what Relational Coding entails.

Conclusion

Aiming to enhance the transparency and trustworthiness of GT studies, I put forward Relational Coding. Acknowledging GT's various iterations and the numerous interpretations of each approach, Relational Coding constitutes a potentially modifiable toolkit, clarifying how and when to record key coding decisions. Naturally, the decisions themselves might be easily modifiable depending on the questions the researcher poses to the data. This tool constitutes a novel approach to recording the process of discovering theory through GT, not only building on numerous existing approaches ranging from manual to digital, rooted in various underlying epistemologies and ontologies, but addresses a gap in the literature. While GT is constantly evolving as scholars continue to shed light on its inner workings by publishing their approaches, demonstrating how the relationships between codes can be captured and made visible beyond the mind of the researcher has yet to be established. Here, Relational Coding provides a useful and inherently modifiable tool for doing exactly that.

Beyond a simple how-to-code guide for GT, Relational Coding aims to demonstrate how to code so that the theory produced by this methodology is trustworthy due to the replicability of the manner in which the results are arrived at, rather than the replicability of the results themselves. While it does not aim to function as a one-size-fits all fix for GT's weaknesses, if applied correctly it will assist not only the reader but also the researcher with regard to a clear and coherent idea of theoretical concept building. In this manner open, intermediate, and theoretical coding can be recording in a manner reflective not only of their outcomes, but their process too. Subsequently, outcomes can be regarded as increasingly reliable and trustworthy due to the transparency of the process behind them. This should allow researchers and their audience alike to trace the emergence of theory from data, to the codes grounded in that data, to the relationships between those codes as they begin from open, narrowed down to intermediate based on relationships between open codes, and finally narrowed down further to conceptually rich theory based on relationships between intermediate codes and their component open codes. However, two key points for improvement remain. First, demonstrating how to modify Relational Coding for methodologically sound application to other strands of GT remains necessary. Second, as noted, differing strands of GT adopt diverging stances on literature reviews. Future iterations of the Relational Coding tool must incorporate clear instruction and demonstration in order to facilitate clarity around the influence of literature on coding. Third, while the Glaserian-inspired classical brand of GT which I adhere to calls for late-stage literature reviews it does not call for an 'empty head' (Richardson and Kramer, 2006). No approach to and subsequent account of coding can hope to be complete and comprehensible without demonstrating the positionality of the researcher, certainly not Relational Coding at the very least. Per Subramani (2019),

whose work on reflexivity in theory building research could usefully be brought to weigh on Relational Coding, the researcher's paradigmatic underpinnings are of great importance. Building transparency around positionality and personal subjectivities into this tool for transparency in more generalised coding remains a key future step for enhancing Relational Coding's usefulness and applicability.

Acknowledgements

I would like to acknowledge the patient efforts of [details omitted for double-anonymised peer review] Dr Jonathan Cane and Dr Caryn Abrahams who provided constant reminders on the importance of transparency in writing up my research over the years. Additionally, the writing mentorship received from Prof Mfaniseni Sihlongonyan and Prof Marie Huchzermeyer over the course of my postdoctoral fellowship was invaluable.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Life in the City doctoral scholarship as administered by the Wits School of Governance, and the Wits-TUB-UNILAG Urban Lab postdoctoral fellowship funded by the German Government through the German Academic Exchange Service (DAAD) and Federal Ministry for Economic Cooperation and Development (BMZ).

ORCID iD

Carmel Rawhani  <https://orcid.org/0000-0002-2767-1889>

Note

1. This section draws from a previously unpublished PhD (anonymised).

References

- Aguinis H and Solarino AM (2019) Transparency and replicability in qualitative research: The case of interviews with elite informants. *Strategic Management Journal* 40(8): 1291–1315.
- Aldiabat K and Le Navenec C-L (2018) Data saturation: The mysterious step in grounded theory method. *The Qualitative Report* 23(1): 245–261.
- Allan G (2003) A critique of using grounded theory as a research method. *Electronic Journal of Business Research Methods* 2(1): 37–46.
- Allen L (2014) A critique of four grounded theory texts. *The Qualitative Report* 15(6): 1606–1620.
- Barlett D and Payne S (2004) Grounded theory—Its basis, rationale and procedures. In: McKenzie G, Powell J and Usher R (eds) *Understanding Social Research Perspectives on Methodology and Practice*. London: Routledge Falmer, pp.173–195.
- Birks M, Hoare K and Mills J (2019) Grounded theory: The FAQs. *International Journal of Qualitative Methods* 18: 1609406919882535.

- Birks M and Mills J (2010) *Grounded Theory: A Practical Guide*. Thousand Oaks, CA: Sage. https://books.google.co.za/books?id=fI0sRC15B6cC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Bowen GA (2009) Supporting a grounded theory with an audit trail: An illustration. *International Journal of Social Research Methodology* 12(4): 305–316.
- Braun V and Clarke V (2021) Can I use ta? Should I use ta? Should I *not* use ta? Comparing reflexive thematic analysis and other pattern-based qualitative analytic approaches. *Counselling and Psychotherapy Research* 21(1): 37–47.
- Bryant A (2017) *Grounded Theory and Grounded Theorizing: Pragmatism in Research Practice*. New York, NY: Oxford University Press.
- Bryant A and Charmaz K (2007) Introduction. In: Charmaz K and Bryant A (eds) *The Sage Handbook of Grounded Theory*. Thousand Oaks, CA: SAGE, pp.1–28.
- Bryant A and Charmaz K (2019) *The SAGE Handbook of Current Developments in Grounded Theory*. Thousand Oaks, CA: SAGE.
- Bryant CA (2002) Re-grounding grounded theory. *Journal of Information Technology Theory and Application* 4(1): 25–42.
- Charmaz K (1996) Rethinking methods in psychology. In: Smith JA, Harre' R and van Langenhove L (eds) *Rethinking Methods in Psychology*. Thousand Oaks, CA: SAGE, pp.27–49.
- Charmaz K (2008) Grounded theory as an emergent method. In: Hessi-Biber SN and Leavy P (eds) *Handbook of Emergent Methods*. New York, NY: The Guildford Press, pp.155–172.
- Charmaz K (2011) Grounded theory methods. In: Denzin NK and Lincoln YS (eds) *The Sage Handbook of Qualitative Research*. Thousand Oaks, CA: SAGE, pp.359–380.
- Charmaz K (2014) *Constructing Grounded Theory*, 2nd edn. Thousand Oaks, CA: SAGE, <https://uk.sagepub.com/en-gb/afr/constructing-grounded-theory/book235960>
- Charmaz K (2017) Special invited paper: Continuities, contradictions, and critical inquiry in grounded theory. *International Journal of Qualitative Methods* 16(1): 160940691771935.
- Charmaz K and Belgrave LL (2019) Thinking about data with grounded theory. *Qualitative Inquiry* 25(8): 743–753.
- Charmaz K and Thornberg R (2021) The pursuit of quality in grounded theory. *Qualitative Research in Psychology* 18: 305–327.
- Chun Tie Y, Birks M and Francis K (2019) Grounded theory research: A design framework for novice researchers. *Sage Open Medicine* 7: 2050312118822927.
- Clarke A (2019) Situating grounded theory and situational analysis in interpretive qualitative inquiry. In: Bryant A and Charmaz K (eds) *The SAGE Handbook of Current Developments in Grounded Theory*. Thousand Oaks, CA: SAGE, pp.3–47.
- Clarke AE (2003) Situational analyses: Grounded theory mapping after the postmodern turn. *Symbolic Interaction* 26(4): 553–576.
- Clarke AE (2007) Grounded theory: Critiques, debates, and situational analysis. In: Outhwaite W and Turner S (eds) *The SAGE Handbook of Social Science Methodology*. Thousand Oaks, CA: SAGE Publications Ltd, pp.423–442.
- Connelly LM (2016) Trustworthiness in qualitative research. *Understanding Research* 25(6): 435–436.
- Corbin J and Strauss A (2008) Techniques and procedures for developing grounded theory. In: Corbin J and Strauss A (eds) *Basics of Qualitative Research*, 3rd edn. Thousand Oaks, CA: SAGE Publications, Inc. <https://doi.org/10.4135/9781452230153>.
- Corbin JM and Strauss A (1990) Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology* 13(1): 3–21.
- Detering NM and Waters MC (2021) Flexible coding of In-depth interviews: A twenty-first-century approach. *Sociological Methods & Research* 50(2): 708–739.
- Dunne C and Üstündağ BG (2020) Successfully managing the literature review and write-up process when using grounded theory methodology—A dialogue in exploration. *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research* 21(1): 25.
- Flick U (2018) *Doing Grounded Theory*. Thousand Oaks, CA: SAGE Publications Ltd.
- Glaser BG (1978) *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory*. Mill Valley, CA, Sociology Press.
- Glaser BG (1992) *Emergence vs. Forcing: Basics of Grounded Theory Analysis*. Mill Valley, CA, Sociology Press.
- Glaser BG (1998) *Doing Grounded Theory: Issues and Discussions*. Mill Valley, CA, Sociology Press.
- Glaser BG (2002a) Constructivist grounded theory? *Forum: Qualitative Social Research* 3(3): 1–14. <http://www.qualitative-research.net/index.php/fqs/rt/prINTERfriendly/825/1792>
- Glaser BG (2002b) Conceptualization: On theory and theorizing using grounded theory. *International Journal of Qualitative Methods* 1(2): 23–38.
- Glaser BG and Holton J (2004) Remodeling grounded theory. *Forum: Qualitative Social Research* 5(2): 1–22. <http://www.qualitative-research.net/index.php/fqs/rt/prINTERfriendly/607/1315>
- Glaser BG (2007) Doing formal theory. In: Bryant A and Charmaz K (eds) *The Sage Handbook of Grounded Theory*. Thousand Oaks, CA: SAGE, pp.97–113.
- Glaser BG (2012) Stop! Write. Writing grounded theory. *The Grounded Theory Review* 11(1): 2–11.
- Glaser BG (2014) Applying grounded theory. *The Grounded Theory Review* 13(1): 46–50.
- Glaser BG (2016) Grounded description: No no. *The Grounded Theory Review* 15(2): 3–7.
- Glaser BG (2019) Grounded description: No no. In: Bryant A and Charmaz K (eds) *The SAGE Handbook of Current Developments in Grounded Theory*. Thousand Oaks, CA: SAGE, pp.441–445.
- Glaser BG and Holton J (2005) Staying open: The use of theoretical codes in grounded theory. *The Grounded Theory Review* 5(1): 1–20.
- Glaser BG and Holton JA (2007) Remodeling grounded theory. *Historical Social Research / Historische Sozialforschung* 19: 47–68.
- Glaser BG and Strauss A (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research*. London: Aldine Publishing Company.
- Guetterman TC, Babchuk WA, Howell Smith MC, et al. (2019) Contemporary approaches to mixed methods—grounded theory research: A field-based analysis. *Journal of Mixed Methods Research* 13(2): 179–195.
- Holton JA (2007) The coding process and its challenges. In: Charmaz K and Bryant A (eds) *The Sage Handbook of Grounded Theory*. Thousand Oaks, CA: SAGE, pp.265–289.
- Holton JA (2010) The coding process and its challenges. *The Grounded Theory Review* 9(1): 21–38.

- Howard-Payne L (2016) Glaser or Strauss? Considerations for selecting a grounded theory study. *South African Journal of Psychology* 46(1): 50–62.
- Howard SD, Lee KL, Nathan AG, et al. (2019) Healthcare experiences of transgender people of Color. *Journal of General Internal Medicine* 34(10): 2068–2074.
- Jacobs AM, Buthe T, Arjona AM, et al. (2019) *Transparency in Qualitative Research: An Overview of Key Findings and Recommendations (Introduction to Working Paper Series: Qualitative Transparency Deliberations, Working Group Final Reports)* (SSRN Scholarly Paper ID 3430025). Social Science Research Network. <https://doi.org/10.2139/ssrn.3430025>
- Keene DE, Guo M and Murillo S (2018) “That wasn’t really a place to worry about diabetes”: Housing access and diabetes self-management among low-income adults. *Social Science & Medicine* 197: 71–77.
- Kelle U (2007) ‘Emergence’ vs. ‘forcing’ of empirical data? A crucial problem of ‘grounded theory’ reconsidered. *Historical Social Research/Historische Sozialforschung* 19: 133–156.
- Kolb SM (2012) Grounded theory and the constant comparative method: Valid research strategies for educators. *Journal of Emerging Trends in Educational Research and Policy Studies* 3(1): 83–86.
- Li X, Du J and Long H (2019) Green development behavior and performance of industrial enterprises based on grounded theory study: Evidence from China. *Sustainability* 11(15): 4133.
- Lopez-Gonzalez H, Estévez A and Griffiths MD (2018) Controlling the illusion of control: A grounded theory of sports betting advertising in the UK. *International Gambling Studies* 18(1): 39–55.
- Maher C, Hadfield M, Hutchings M, et al. (2018) Ensuring rigor in qualitative data analysis: A design research approach to coding combining NVivo with traditional material methods. *International Journal of Qualitative Methods* 17(1): 160940691878636.
- Miguel E, Camerer C, Casey K, et al. (2014) Promoting transparency in Social Science Research. *Science* 343(6166): 30–31.
- Moravcsik A (2014) Transparency: The revolution in qualitative research. *PS: Political Science & Politics* 47(1): 48–53.
- Moravcsik A (2020) *Transparency in qualitative research*. Thousand Oaks, CA: SAGE Publications Ltd. <https://doi.org/10.4135/9781526421036>
- Nathaniel A (2020) How to read classic grounded theory. *Grounded Theory Review* 19(9): 1–7.
- Nathaniel A (2021) Classic grounded theory: What it is and what it is not. *Grounded Theory Review* 20(1): 1–7.
- Nelson LK (2020) Computational grounded theory: A methodological framework. *Sociological Methods & Research* 49(1): 3–42.
- Reichertz J (2007) Abduction: The logic of discovery in grounded theory. In: Bryant A and Charmaz K (eds) *The Sage Handbook of Grounded Theory*. Thousand Oaks, CA: SAGE, pp.214–228.
- Reichertz J (2019) Abduction: The logic of discovery of grounded theory – An updated review. In: Bryant A and Charmaz K (eds) *The SAGE Handbook of Current Developments in Grounded Theory*. Thousand Oaks, CA: SAGE, pp.259–281.
- Richardson R and Kramer EH (2006) Abduction as the type of inference that characterizes the development of a grounded theory. *Qualitative Research* 6(4): 497–513.
- Rieger KL (2019) Discriminating among grounded theory approaches. *Nursing Inquiry* 26(1): e12261.
- Sbaraini A, Carter SM, Evans RW, et al. (2011) How to do a grounded theory study: A worked example of a study of dental practices. *BMC Medical Research Methodology* 11(1): 128.
- Sikolia D, Biros D, Mason M, et al. (2013) Trustworthiness of grounded theory methodology research in information systems. *MWAIS 2013 Proceedings*. <https://aisel.aisnet.org/mwais2013/16>
- Sinclair S, Hack TF, Raffin-Bouchal S, et al. (2018) What are healthcare providers’ understandings and experiences of compassion? The healthcare compassion model: A grounded theory study of healthcare providers in Canada. *BMJ Open* 8(3): e019701.
- Strauss A and Corbin J (1994) Grounded theory methodology: An overview. In: Denzin NK and Lincoln YS (eds) *Handbook of Qualitative Research*. Thousand Oaks, CA: SAGE, pp.273–285. <http://psycnet.apa.org/record/1994-98625-016>
- Strauss A and Corbin J (1998) *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2nd edn. Thousand Oaks, CA: SAGE.
- Strauss AL (1987) *Qualitative Analysis for Social Scientists*. Cambridge: Cambridge University Press.
- Subramanian L, Kirk R, Cuttitta T, et al. (2019) Remote management for Peritoneal Dialysis: A qualitative study of patient, care partner, and clinician perceptions and priorities in the United States and the United Kingdom. *Kidney Medicine* 1(6): 354–365.
- Subramani S (2019) Practising reflexivity: Ethics, methodology and theory construction. *Methodological Innovations* 12(2): 2059799119863276.
- Timonen V, Foley G and Conlon C (2018) Challenges when using grounded theory: A pragmatic introduction to doing GT research. *International Journal of Qualitative Methods* 17(1): 160940691875808.
- Tuval-Mashiach R (2017) Raising the curtain: The importance of transparency in qualitative research. *Qualitative Psychology* 4(2): 126–138.
- Williams EN and Morrow SL (2009) Achieving trustworthiness in qualitative research: A pan-paradigmatic perspective. *Psychotherapy Research* 19(4–5): 576–582.
- Wong TKM, Man SS and Chan AHS (2020) Critical factors for the use or non-use of personal protective equipment amongst construction workers. *Safety Science* 126: 104663.
- Yu M and M Smith S (2021) Grounded theory: A guide for a new generation of researchers. *International Journal of Doctoral Studies* 16: 553–568.

Author biography

Carmel Rawhani is a Sector Coordinator for the KfW Development Bank’s work in Violence Prevention, Water, and Finance in South Africa. She is also a visiting fellow at the Wits-TUB-UNILAG Urban Lab, based at the Wits School of Architecture and Planning in Johannesburg. Having completed her PhD which focused on social cohesion, public space, and the policies which surround them, Carmel is interested in furthering her contributions to her field through continued work on these themes and the methods which she adopted in studying them. Here, working with grounded theory stands out as one of her core areas of interest.

Appendix A

Table A. Top ten GT articles from 2018 to 2021 on Web of Science, ranked in order of citations.

Author/s	Article title	Year	Contribution to GT practice
Ylona Chun Tie, Melanie Birks and Karen Francis	Grounded theory research: A design framework for novice researchers	2019	The authors usefully summarize the evolution of GT to provide clarity to novices, but beyond clarifying rich descriptive summary there is no new guidance on coding.
Khaldoun M. Aldiabat, Carole-Lynne Le Navenec	Data Saturation: The Mysterious Step in Grounded Theory Methodology	2018	Focuses on coding insofar as it relates to data saturation and demonstrating the achievement of saturation transparently. Much of the guidance is conceptual rather than practically-demonstrative.
Laura K Nelson	Computational Grounded Theory: A Methodological Framework	2020	Provides a new framework for a computational GT approach. This is quite similar to the aim of Relational Coding in terms of enhancing transparency, however it aims for completely reproducible coding, which departs from my approach to GT. While this serves to enhance GT legitimacy it does not demonstrate how to make the relationships between codes clearer, and thus there is a degree of decision-making which is obscured even though overall transparency in other areas is enhanced.
Virpi Timonen, Geraldine Foley, and Catherine Conlon	Challenges When Using Grounded Theory: A Pragmatic Introduction to Doing GT Research	2018	This article focuses more on improving GT by clarifying its underlying concepts. This speaks to the core tenets and philosophy of GT but does not seek to contribute practicable models for transparency specifically.
Kathy Charmaz and Robert Thornber	The pursuit of quality in grounded theory	2020	This paper historicizes the weaknesses of GT and proposes key improvements. While some examples of coding tables are provided, they provide clarity at the level of summary, rather than providing clarity of emergence and the relationships between codes a foundation for theory building.
Kendra L Rieger	Discriminating among grounded theory approaches	2019	This article concerns itself with the task of clarifying the evolution of GT to enable researchers to reach informed decisions when choosing a particular strand to follow.
Shane Sinclair, Thomas F Hack, Shelley Raffin-Bouchal, Susan McClement, Kelli Stajduhar, Pavneet Singh, Neil A Hagen, Aynharan Sinnarajah, Harvey Max Chochinov	What are healthcare providers' understandings and experiences of compassion? The healthcare compassion model: a grounded theory study of healthcare providers in Canada	2018)	This study produces a grounded theory which seeks to map out the dimensions of compassion on the part of Canadian healthcare providers. While it provides a useful example of how the grounded theory itself might be summed up diagrammatically, and explains coding conducting by a group of researchers in depth, it does not demonstrate exactly how the codes emerged, or how the different stages of coding contributed to each other (as Relational Coding does).
Hibai Lopez-Gonzalez, Ana Estévez & Mark D. Griffiths	Controlling the illusion of control: a grounded theory of sports betting advertising in the UK	2018	This paper provides a grounded theory of the narratives which normalises gambling in the form of sports betting and the related advertising industry. While a degree of transparency is provided by summarising the codes in a table which cites their frequency of occurrence within the data, it isn't clear how the various rounds of coding lead to one another.
Xingwei Li, Jianguo Du, and Hongyu Long	Green Development Behavior and Performance of Industrial Enterprises Based on Grounded Theory Study: Evidence from China	2019	This paper develops a grounded theory of green development behaviour and performance in industrial enterprises to explain the drivers thereof. While the article provides transparency in terms of clearly demonstrating how the research design unfolds, with clarity on the various coding phases, it is not clear how exactly each phase informs the next and what the relations between the codes are.
Timothy C. Guetterman, Wayne A. Babchuk, Michelle C. Howell Smith, and Jared Stevens	Contemporary Approaches to Mixed Methods–Grounded Theory Research: A Field-Based Analysis	2019	This paper provides useful practicable guidelines for adopting a mixed methods approach to GT, but beyond this macro-level focus it does not zoom in on coding. This is framed as a contribution to enhancing transparency in GT, but focuses on a different area to that of Relational Coding.

Appendix B

Table B. Top ten GT articles from the last 5 years on Web of Science mentioning coding and grounded theory in the topic.

Author/s	Article title	Year	Contribution to GT practice
Nicole M. Deterding and Mary C. Waters	Flexible Coding of In-depth Interviews: A 21st-century Approach	2021	The article argues that GTM is an obstacle to research transparency and that it is out of step with modern interview approaches, with a particular focus on coding software. Their contribution is to advocate for the opposite of what grounded theorists do. They acknowledge that demonstrating not only the grounded quality of codes, but also their relationships is key, but do not transparently demonstrate how to do so (though they do demonstrate how else to use qualitative coding software).
Carmel Maher, Mark Hadfield, Maggie Hutchings, and Adam de Eyto	Ensuring Rigor in Qualitative Data Analysis: A Design Research Approach to Coding Combining NVivo With Traditional Material Methods	2018	The paper supports the notion that traditional coding tools (writing on paper) and modern software (NVivo) can suitably support rigorous coding. The researchers achieve great transparency using basic tools such as sticky notes, highlighters, and hand-drawn diagrams. They do not, however, demonstrate how relationships between codes produced their outcomes.
Virginia Braun and Victoria Clarke	Can I use TA? Should I use TA? Should I not use TA? Comparing reflexive thematic analysis and other pattern-based qualitative analytic approaches	2021	This paper provides an overview of many types of GT coding alongside one another to contrast it with other coding-heavy analytic approaches. It discusses the importance of research rigour but doesn't show how that can be demonstrated by the researcher when it comes to coding relationships.
Esmita Charani, Ingrid Smith, Brita Skodvin, Anne Perozziello, Jean-Christophe Lucet, François-Xavier Lescure, Gabriel Birgand, Armel Poda, Raheelah Ahmad, Sanjeev Singh, Alison Helen Holmes	Investigating the cultural and contextual determinants of antimicrobial stewardship programmes across low-, middle- and high-income countries – A qualitative study	2019	This grounded study achieves transparency in explaining themes, demonstrating a degree of groundedness, and analysis as it builds. However it does not demonstrate exactly how coding occurred, lead to further coding, and resulted in theory that draws not only on data but the relations between the codes emergent therefrom.
Elizabeth A. Samuels, Chantal Tape, Naomi Garber, Sarah Bowman, Esther K. Choo	'Sometimes You Feel Like the Freak Show': A Qualitative Assessment of Emergency Care Experiences Among Transgender and Gender-Nonconforming Patients	2017	This study demonstrates how the team located codes, but not how the relations between codes drove progress from one phase of coding to the other, thereby inadvertently obscuring part of the theory.
Susanna D. Howard, Kevin J. Lee, Aviva G. Nathan, Hannah C. Wenger, Marshall H. Chin, Scott C. Cook	Healthcare Experiences of Transgender People of Color	2019	Examples of quotes from which codes arose are shared, but it is not clear how the codes produce the final outcomes, and it is not easy to discern how the related quality of the codes comes to bear on the final theory.
Tom Ka Man Wong, Siu Shing Man, Alan Hoi Shou Chan	Critical factors for the use or non-use of personal protective equipment amongst construction workers	2020	The paper demonstrates an appreciation for the importance of the relationships between codes but demonstrates the end result of coding rather than the result as an outcome of a process guided by the relationships between codes. There is not clear demonstration in the model used by these authors of a method through which to show the relations between codes and their impact on final theory.
Lalita Subramanian, Rosalind Kirk, Tony Cuttitta, Nicole Bryant, Kimberly Fox, Margie McCall, Erica Perry, June Swartz, Yanko Restovic, Allison Jeter, Angelito Bernardo, Bruce Robinson, Jeffrey Perl, Ronald Pisoni, and Rachel L. Perlman	Remote Management for Peritoneal Dialysis: A Qualitative Study of Patient, Care Partner, and Clinician Perceptions and Priorities in the United States and the United Kingdom	2019	The paper uses grounded theory coding approaches but does not result in a grounded theory. Rather, it results in rich descriptive thematic analysis. An approach to teasing out the relationships between codes is absent.
Lauren J. Breen, Samar M. Aoun, Moira O'Connor, Denise Howting, and Georgia K.B. Halkett	Family Caregivers' Preparations for Death: A Qualitative Analysis	2018	The study mentions that coding took place per GT methods and that diagramming was key to developing and understanding the relationships between codes throughout the process, but this is not demonstrated beyond written descriptions in a few sentences.
Danya E. Keene, Monica Guoa, Sascha Murillo	'That wasn't really a place to worry about diabetes': Housing access and diabetes self-management among low-income adults	2018	The article confirms that coding took place per GT guidelines but does not demonstrate the coding. Instead, a short description is provided before moving on to the findings.

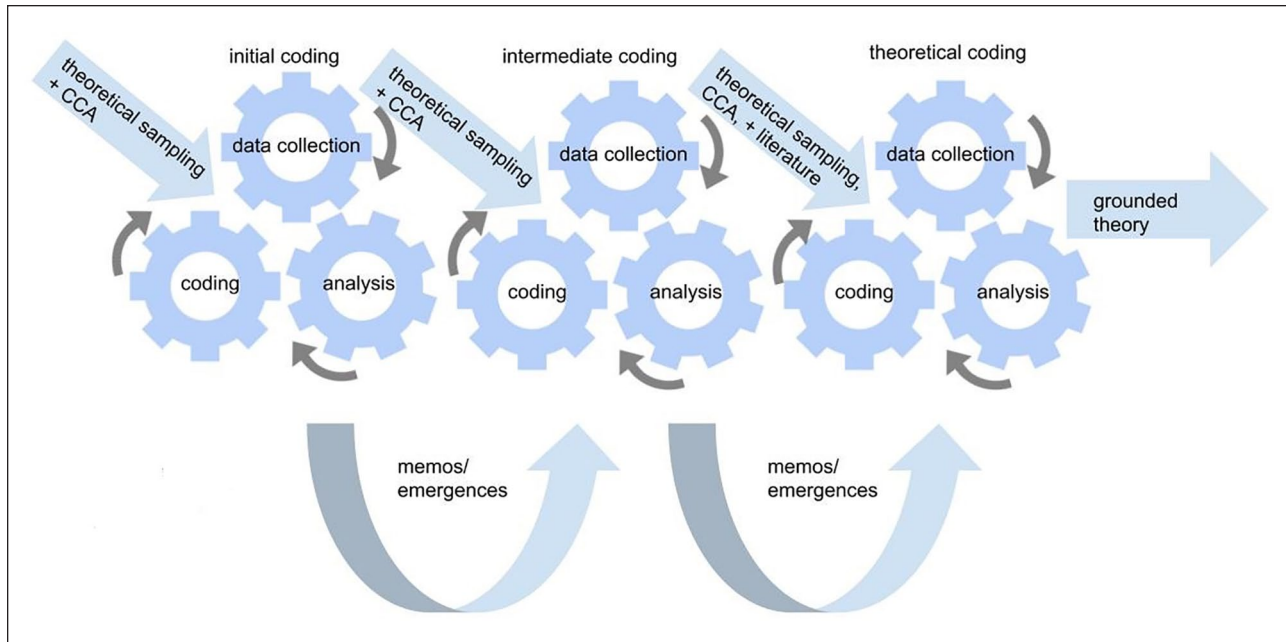


Figure A. Grounded theory, summarised.

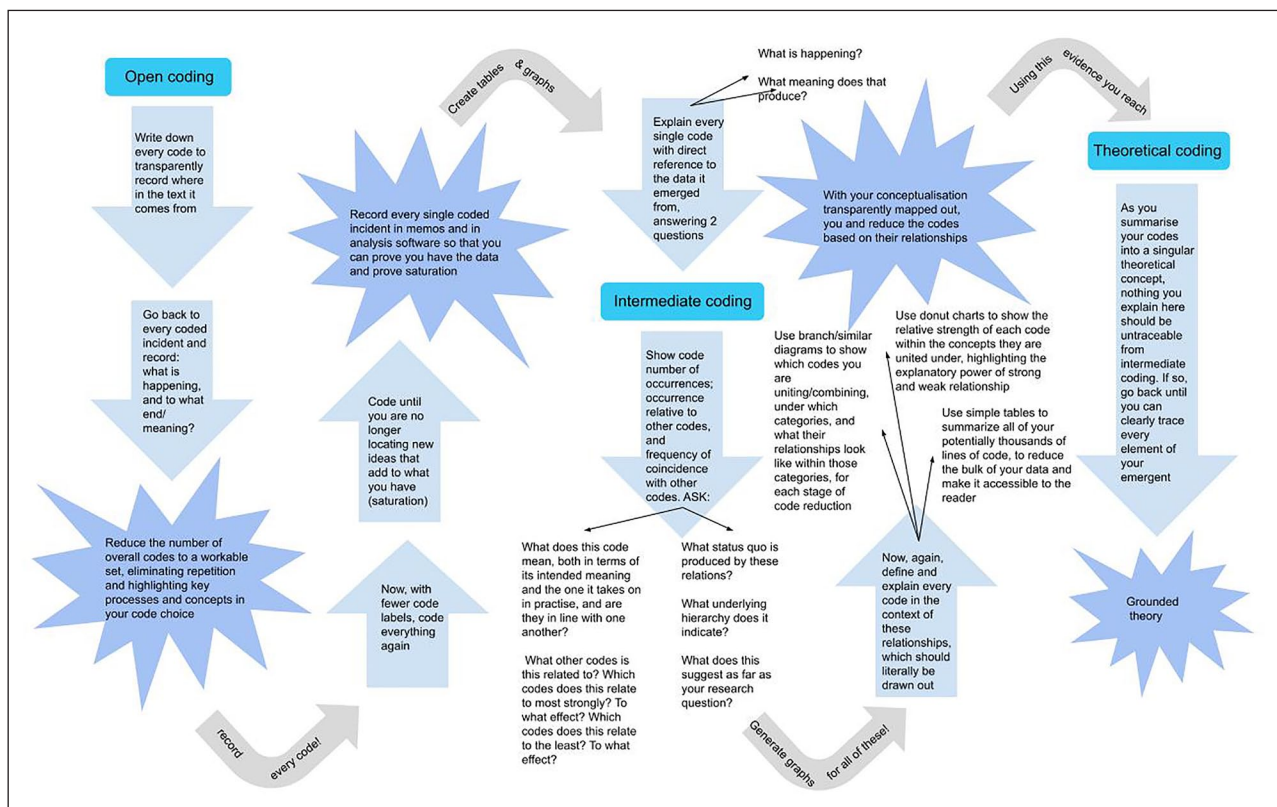


Figure B. A summary of Relational Coding.