2. How Readers Recognize Meaning

Have you ever seen those barcode readers in the stores? Some are hand held and others are built-in under a glass plate at the checkout stand. The barcode on an item to be purchased is swiped over the reader. The reader scans the code and recognizes the combination of bars of varying thickness. It then matches that barcode to the numerical price of the item and sends a signal for that price to be projected in digital form on the cash register display.

Human readers of language code are somewhat like that. The language code is “swept” in front of the windows of the eyes. The eyes scan the code and recognize the combinations of letters of various shapes. They match those letter combinations to the expanded code of each item stored in memory, and then send a signal for that expanded code to be projected on the display surfaces of the brain.

The point is this: In order for a transfer of information to take place from the code to the display, there is a piece of equipment involved that has to be working properly: the code reader. An accurate code reader is just as important for an accurate transfer of information as is an accurate text of the code. A defective code reader will not recognize code correctly. It will produce inaccurate signals, and the unavoidable result is a display projecting false data. In other words, it does people little good to swipe the code of a writing—including the scriptures—in front of their eyes if their code reader is not functioning properly.

The development of accurate readers

It’s instructive to look at the standard way in which readers are produced. First, the reader is created with a built-in scanner, a central processing unit, various interfacing instruments, and plenty of memory. Most of this equipment is better known as eyes, ears, brains, and assorted body parts. These are human babies, the future readers of language code. Though marvelous little gadgets, they are delivered only partially programmed for reading.

The character-recognition software program that actually turns babies into code readers requires years to install and learn. The most common processes that have to be initiated are *listening to people talk; recognizing sounds, words, and word combinations; learning to talk; listening to stories; going to school; learning the letters; learning to read; learning to write; doing homework; going to more schools; doing more homework; and getting plenty of daily practice.*

Some of this is learning of the formal kind but most of it isn’t. The developing code readers adjust and test their reading skills as they grow older by regularly reading everything from the signs and billboards out on the streets to magazines, newspapers, bank printouts, bills, e-mails, books, and the many other texts of language code that become a part of their lives and work.

The developing readers learn to read the very same codes and codings procedures we discussed earlier: letters, words, phrases, clauses, sentences, paragraphs, chapters, and whole pieces of literature. Along with the codes, they also learn the coding procedures: situation coding, genre coding, prominence coding, reference coding, rhetoric coding, relation coding, and form coding. None of the codes and coding procedures are learned separately or one after the other. They are learned all together. Even toddlers learn all the coding procedures. They learn quickly, for example, that meaning is dependent on a situation. They observe that Mommy talks differently than Daddy; they notice that it's not the same if Grandpa says something or if big brother says it; they recognize that bedtime talk is different than shopping talk.

Toddlers also learn genre coding**.** They quickly become acquainted with fairy tales, songs, poems, bedtime stories, jokes, adventure books, and birthday cards. They observe how Mommy and Daddy read books and other written materials, and they hear the names of them. Toddlers learn very quickly that a story has a beginning, middle, and end. They won't let you begin a story in the middle, and they insist on reading to the very end.

Think of the big role played by prominence coding, reference coding, and rhetoric coding in the life of a toddler. Small children recognize what a story is about (prominence coding), they ask what words mean (reference coding), and they get great delight from the nonstandard expressions used in their children's books (rhetoric coding).

Small children also do not neglect learning relation coding and form coding. One of the amazing things about two and three-year olds is that they can often be heard using word endings and sentence structures perfectly. By the time they are four they seldom make basic relation-code or form-code mistakes.

The way children learn language codes sets the stage for years of language learning which grows increasingly more complex. As time passes—through grade school, high school, college, occupational training—each growing code reader eventually begins to specialize in reading certain kinds of code such as medical code, mechanical code, mathematical code, and the codes of computer, business, law, geography, aviation, management, agriculture, etc. Some code readers develop a mastery of huge networks of language code in a particular area of knowledge and become recognized as experts.

It's important to note, however, that a code reader who is very good in one area of language code, even expert, is not necessarily a good reader in a different subject area of language code. A professor of mathematics, for example, may be a recognized authority in statistics, and yet have little more understanding than a child in a middle school when reading texts dealing with auto repair, cooking, soccer, or archaeology.

Another observation we can make is that readers must have certain extensive networks of code filed in their memories in order to understand the small codes that are details of those networks.

Consider, for example, number code. Though a person may read a series of numbers in a text, it is not enough to have learned only those numbers. The entire network of code for numbers needs to be in one's memory because number code can only be sufficiently understood as a network of code. Numbers are actually parts of various networks including systems such as sequences (1,2,3, . . .), decimal combinations (4,329.24), functions (adding, subtracting), fractions (one-half), formulas ($a^2 + b^2 = c^2$), years (1945), and many more. All these networks and the code systems that make them work have to be learned, however, before any of the individual numbers make any sense to a reader.

Here are more examples of large networks of code that have to be mastered in order to understand the details that are a part of the network:

- Time/calendar system of seconds, minutes, hours, days, weeks, months, years, etc.
- Search systems based on alphabetizing and outlining
- Charting and graphing
- Financial transactions and bookkeeping
- Addresses and telephone numbers
- Traffic rules and street signs
- The rules of sports, board games, and card games
- Computer operating systems

This list is only the tip of the iceberg when it comes to the networks of language code that need to be understood to recognize simple codes such as *three O'clock, the third street, the top twenty percent, 304-567-2198, sixty miles per hour, Go directly to jail. Do not pass Go., $4.50,* and *open a blank document.* In fact, the knowledge of several large networks of code is usually necessary to understand most pieces of information. A simple bank printout, for example, combines numerical code, calendar code, table code, alphabetic code, and address code.

Only informed readers are accurate readers

It should be clear now that any individual language code can only be understood when it is seen as a part of a much larger network of code. This is why all school systems from the first grade all the way through university have to be sure that the students learn language code in a correct sequence. Learning numbers, for example, precedes learning adding and subtracting, which precedes learning fractions, which in turn precedes learning algebra, and so on. In all schools, many of the courses are considered prerequisites for other courses. The prerequisite courses establish networks of code in the minds of the students which are necessary before other subjects can be learned.

The purpose of these observations is to emphasize that *code readers must have sufficient networks of corresponding code firmly stored in their memory to be able to correctly read the language code of any given text.* The implications of this fact should be considered by anyone who wants to become an accurate reader of the writings penned by the students of Jesus Christ. Like all writers, they used the language code corresponding to the networks of code in their own minds and to those networks which were also in the minds of their intended readers. To understand the code of those original writers and readers, the present-day reader has to learn the prerequisite networks of code understood by the original writers and readers of the code.

The process of reading

An accurate reader observes all the language codes in a written text, notes the role of each individual code, watches how each code works together with other codes to construct larger units of code, and observes how the larger units of code ultimately combine to form a single network of code. The reader thoroughly understands the text when the main prominence of the whole network of code is recognized and when the subprominent units of code are correctly related to that main prominence.

You might want to read that again. In simpler words, it's merely saying that a reader has to build the whole from the parts and then define the parts by the whole.

Let's say that two mechanics are each presented with a large box full of parts—bolts, screws, pistons, gaskets, belts, cables, batteries, wheels, tires, and hundreds of other assorted items. Each box contains an unassembled machine. At first it is not clear what sort of machine each box of parts represents. Many of the parts look very similar.

The mechanics begin to assemble the smaller units that they recognize—a carburetor, a generator, an axle, a starter, etc. Then they observe the wheels, the style of seats, the steering devices. Gradually it becomes clear that one mechanic has the parts of a small motorcycle and the other has the parts of a riding lawnmower.

In this example, the "main prominence" of the parts was recognized when it became clear what all the parts were designed to create. The one set of parts became a network of parts called a motorcycle. The other became a network known as a riding lawnmower.

The mechanics were able to figure out each of the machines because they were trained to "read" machine language. They knew how small mechanical parts fit together; they easily recognized the smaller components and knew the roles they play in larger mechanical combinations. Certain parts stood out—the handlebars of the motorcycle and the blades of the mower—because they were unique to particular kinds of machines. It was these easily recognizable parts that made the identity of the whole machine quickly evident.

Had one of the machines been something the mechanics had never seen, however, then they may have never discovered what it was. It's hard to assemble something you've never seen. In the same way, it is difficult to understand a particular genre of writing that you've never read before.

This is how the reading of a text takes place. From the parts, a meaningful whole has to be recognized. Like the mechanics, the reader has to be able to recognize the smallest parts, know how they fit together, and be able to recognize the larger units and the roles that they play. These ever larger units in a writing are not batteries and carburetors, but phrases, clauses, sentences, paragraphs, chapters, and the complete piece of literature.

A closer look at the process of reading

We'll begin by observing a simple reading situation and analyze how a reader goes about understanding the coding of the text:

> John Smith comes home, and notices a letter in the mailbox. He grabs it and takes a quick look at the address on the front.
>
> *Mr. and Mrs. John Smith*
> *346 Emory St.*
> *Dayton, OH 43054*
>
> He steps into the house, sees his wife, Helen, and says, "Look what just came in the mail. Some sort of invitation."
>
> "Who's it from?" she asks.
>
> He removes a folded card and reads the following text:.
>
> *We've been married twenty wonderful years.*
> *Join us at our home, Saturday, June 14, at 7 pm to celebrate.*
> *Just bring yourselves and your appetites. No gifts.*
>
> *Fred and Emily Mortenson*
> *RSVP: 806-433-6584*
>
> "It looks like Fred and Emily are having a wedding anniversary. We need to tell them if we are coming or not."

John's remark shows that he understood the message. How? He did it by observing the codes of the writing in two distinct phases. In the first phase, he oriented himself to the writing by making a series of initial observations about the coding of the text.

John's initial observations before opening the envelope:

- Situation code: The writing was in the mailbox at his home. It was in an envelope and had arrived that day.
- Genre code: The writing was some sort of letter.
- Prominence code: He and his wife were the intended receivers of the text.
- Reference code: *Mr. and Mrs.* means *both husband and wife*
- Rhetoric code: The address on the envelope was formal.
- Relation code: The address had three lines with the names, street address, town, and zip in a standard order.
- Form code: The address was handwritten.

A reader's initial observations of the codes of a text pertain to the circumstances of the writing, the language of the writing, and the language codes which are the most obvious. Note, too, that the initial observations included all seven of the language codes. Though these observations are often so automatic that readers seldom take notice of them, they nevertheless provide a great deal of information. What if, for example, the letter had been found in an old chest in the attic instead of in the mailbox? Or what if the names on the front were not those of John and Helen? Would John have opened the letter?

After his initial observations of the codes, John then turned to the second phase which was a complete and detailed reading of the text. His observations once again included all seven kinds of language coding. Following is a selection of the codes he observed in the text:

> *Situation codes:* The specific situation codes include a date, two married couples (authors and receivers), an occasion, and a place.
>
> *Genre codes:* The document is a personal invitation.
>
> Prominence codes: The topic of the text is a party. The purpose of the writing is to extend an invitation.
>
> *Reference codes:* Special references which John had to deal with include the phrase *married 20 years* and the abbreviation *RSVP*.
>
> > Isn't it interesting that John knew the party was an anniversary party even though the word anniversary was never used? This is because the phrase *married 20 years* is culturally understood as a milestone period of time that is often celebrated by married couples.
> >
> > *RSVP* is an abbreviation of a French sentence which has come to have a special usage in English. It is included at the end of invitations requesting that the invited person inform the host whether he or she will be attending the event or not.
>
> *Rhetoric codes:* Two examples include the sentence *Just bring yourselves and your appetite* and the sentence *No gifts*.
>
> > *Bring yourself and your appetite* is a nonstandard sentence because these items are not generally objects of the verb *bring*. People don't normally *bring themselves* or *bring appetites*. The rhetorical coding is designed to set the reader at ease by addressing a potentially embarrassing topic with humor.
> >
> > The phrase *No gifts* is also not a standard sentence. It has no verb. It's a sentence, however, because it borrows the verb from the previous sentence. The word gifts references the cultural practice of giving anniversary gifts. The abruptness of the sentence gives it a certain "sauciness" and also gives it the sense of an imperative (command). It basically says *Please, don't bring any gifts to us.*

Relation codes: Two examples of complex relation codes include the phrase *twenty wonderful years* and the phrase *RSVP: 806-433-6584*

> *Twenty wonderful years* is a case of two consecutive adjectives modifying a noun. The second adjective forms a unit with the noun—*wonderful years*—and the first adjective modifies that unit of code. This is standard English relation coding.
>
> The phrase *RSVP: 806-433-6584* is an example of an abbreviation which communicates a request. The colon (:) is a pointer to the following telephone number, and the entire construction means *Please let us know ahead of time if you will be coming by calling us at 806-433-6584.* It's interesting that this string of code does not have a single code in the form of a word. It is an abbreviation with a punctuation sign and a numeral. It refers, however, to an expanded text of words. (This is a complex case of both relational and referential coding.)

Form codes: Two examples of complex form codes are the word *we've* and the abbreviation *RSVP*.

> *We've* is a combination of the words *we* and *have*. The apostrophe (') replaces the missing letters. Such *contractions* are typical shortened code forms often used in English. Abbreviations such as *RSVP* are another example of shortened code.

All of these examples simply demonstrate how a reader must recognize and accurately record the contribution that each language code makes to a writing. Overlooking a code or inaccurately applying a code will lead to misunderstandings even in a writing as simple as an invitation to a party. For example:

- Situation code: What if John had looked at the *June 14* date and thought it said *July 14*?
- Genre code: What if John had mistaken the letter for an advertisement and just tossed it aside?
- Prominence code: What if John had not deducted from the *bring-your-appetite* phrase that it was going to be a *dinner* party?
- Reference code: What if John had glanced at the first line of the address *Mr. and Mrs. John Smith* and assumed that the letter was for his elderly mother and father?
- Rhetoric code: What if John thought *No gifts* was only meant as a polite phrase?

• Relation code:	What if John failed to notice the RSVP in front of the telephone number?
• Form code:	What if John was not a native speaker of English?

In any one of these cases, the result could have been a misunderstanding with unpleasant consequences.

Though it was a complex task, it probably took John less than a minute to go through both phases of observation with the invitation. Almost automatically, he recognized the individual codes and the total network of code that they formed. He could do this not because it was easy, but because over the years he had become a skilled and experienced reader of this kind of information.

Just think of all the learning John did to become a reader who can deal with code this quickly. He grew up listening to the people around him speak and practicing this speech himself. He also observed them reading and memorized many stories that were read to him. Then he went to school and learned English form codes (letters and spelling), relation codes (phrases and sentences), reference codes (vocabulary), rhetoric codes (exclamations, jokes), prominence codes (subjects, topics), genre coding (stories, math problems, reports, charts, maps, invitations), and became aware of all the many situation codes surrounding a writing (temporal matters, historical particulars, people involved, geographical settings, languages used).

He learned language codes and coding procedures in schools for over 16 years—elementary school, middle school, high school, and the university. It is not an accident that he can look at an invitation and immediately process the language code.

Had a different kind of writing been deposited in his mailbox, John might have had greater difficulty with it. If a request had come from the IRS concerning John's tax returns, for example, it might have contained unfamiliar language code. He may have had to spend several hours working through the text with a dictionary. Regardless of what kind of writing it was, however, he would have had to repeat the two phases of observing the text—the initial phase for orientation and the detailed phase for a comprehensive understanding. In each phase, he would have also had to factor in the contribution of each of the seven kinds of language code and coding procedures.

Understanding information requires an accurate text and an accurate reader

A writing is a precise network of language codes and coding procedures. In order for readers to understand the meaning of a writing, they must recognize and correctly decode all the language codes and coding procedures in the text. There is little room for error. Even slight changes in a portion of a code can change the meaning of an entire text.

Therefore, a translator who has the task of translating a writing for a reader must realize that there are *two* tasks that have to be performed. The first is to ensure that the translated text accurately represents all the language codes of the original document, and the second task is to ensure that the reader of the translated text is equipped to accurately recognize all the language codes of the text. A lack of accuracy of either the text or the reading of the text will result in deficient understanding or in misunderstanding.

Both tasks—ensuring an accurate text and an accurate reading of the text—form the basis for the evaluation of scripture translations that follows. Five selected New Testament translations will be evaluated both in terms of their accuracy in representing the language codes of the original Greek texts and in terms of how well the texts are adapted to actual readers.