

Visual Syntax Diagrams for Programming Language Statements

Lisa M. Braz
Informix Software, Inc.

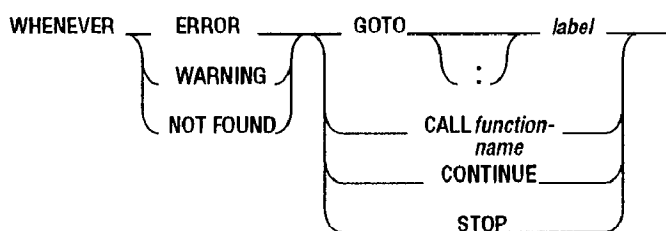
Introduction

In an attempt to help users understand the syntax of programming language statements, some software companies are replacing the traditional Backus-Naur Forms (BNF) diagrams with more visual diagrams called railroad diagrams.

A typical BNF diagram looks as follows:

```
WHENEVER {ERROR | WARNING | NOT FOUND}
          {GOTO [:] label | CALL function-name |
           CONTINUE | STOP}
```

A railroad diagram of the same syntax looks as follows:



According to J. Henno (1987), programming language statements still remain one of the most widely used forms of interaction between the computer and the human user and they have hardly changed since the introduction of Fortran. He says that the presentation of syntax is equally as important [as the design of the programming language], since this is the base for a user to form her or his own model of the language. The presentation mirrors the structure of the language; therefore the language, whose grammar is difficult to understand, is usually also difficult to use.

As technology reaches the masses, through the increase in languages such as fourth generation programming languages (4GL), programmers can be less technically sophisticated than the early Fortran programmers. For example, Gordon

Kerr, Vice-president of MIS for Hyatt Hotels Corp. said that with older systems "You need three to four years of experience to be productive...but only one to two years with a 4GL."

Therefore, more than ever, it is important to help programmers understand the syntax of a programming language.

This paper discusses the following topics:

- The advantages and disadvantages of using railroad diagrams as opposed to BNF diagrams.
- How to determine whether to use railroad diagrams.
- Converting BNF diagrams into railroad diagrams.

The information in this paper is partially based on the results of experiences converting BNF diagrams to railroad diagrams in a 4GL manual for Informix.

Advantages of Using Railroad Diagrams

Railroad diagrams offer the following advantages over BNF diagrams:

- Help users visualize the syntax of a statement.
- Are easier for users to understand.
- Break the statement into logical parts.

Visualization

Railroad diagrams offer a more visual presentation of a statement's syntax than BNF diagrams. Railroad diagrams use visual techniques such as aligning options vertically and giving the user a visual path that she or he can follow.

According to R. John Brockmann (1990), people *reading to do* as opposed to *reading to learn*, benefit from a visual presentation. Research proves, he says, that people reading to do are typically adults reading in a business or commercial environment. Assuming the people reading your manual are reading to do, a more visual presentation is necessary.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

Easy to Use

A group at Informix did a small study on using railroad diagrams before converting our manuals from BNF to railroad diagrams. We tested eight employees with seven or fewer months experience at Informix. The employees were sales representatives, QA specialists, technical writers, and technical editors. Each subject had taken an introductory course in the 4GL language being tested. We tested half of the subjects with BNF diagrams and the other half with railroad diagrams. The test had the subjects write five different formats of a particular statement based on a syntax diagram, some supporting text, and some examples. The tests were exactly the same except for the syntax diagrams. The people with the railroad version of the test scored an average of 90% correct while people with the BNF diagrams scored an average of 55% correct. User's claimed that they used the examples more than the syntax diagrams; however, the big gap in the scores indicates that the subjects probably did use the railroad diagrams to some extent.

Another approach to ensuring the usability of the railroad diagrams is to survey the users of the diagrams. Tandem Computers put railroad diagrams in the *TACL Programmer's Guide*, a programmable command interpreter that provides an interface to an operating system. The writer, Dave Kirby, sent a survey with the initial version of the manual asking for reactions to the diagrams. Over fifty programmers responded. He found that 77% of the respondees found the railroad diagrams easy to understand, and 70% found the diagrams easy to use. Over half of the respondees thought the railroad diagrams should replace the BNF diagrams in other Tandem manuals.

Logical Breakdown

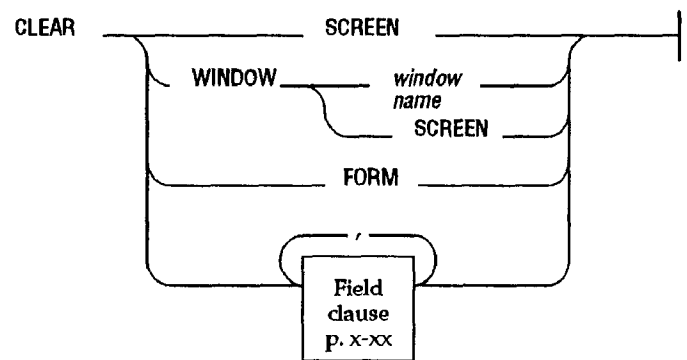
Railroad diagrams logically break down a statement by doing the following:

- Vertically aligning equal options.
- Using sub-diagrams to indicate common tracking between statements and to break down complicated statements.

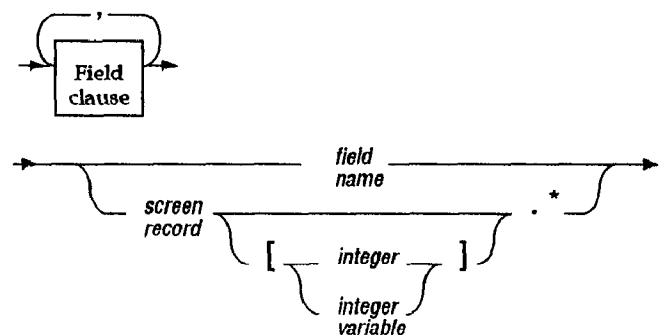
Vertically aligning equal options. In Edward R. Tufte's book, *Envisioning Information*, he suggests methods for escaping the flatland of paper and terminal screens. One method is to use the vertical alignment to add another dimension to a diagram. He suggests using the vertical dimension to array parallel sequences of thoroughgoing data. The railroad diagram of the WHENEVER statement shown at the beginning of this paper demonstrates the effect

of using the vertical. The ERROR, WARNING, and NOT FOUND keywords are equal elements. You need to choose one of these keywords to make a legal statement. In the railroad diagram, these options are stacked. However, in the BNF diagrams, these options are displayed horizontally and seem equal to the WHENEVER keyword and all the other elements.

Using sub-diagrams. Railroad diagrams can further breakdown a statement by using sub-diagrams. Use sub-diagrams to indicate common tracking between multiple statements. For example, the following statement includes a sub-diagram titled Field clause. The Field clause is a section of tracking used by several statements. Therefore, it is broken out into a sub-diagram. The main diagram looks as follows:



The sub-diagram looks as follows:



You will probably want to place the Field clause close to the CLEAR statement. Tufte says "If the visual task is contrast, comparison, and choice—as so often is—then the more relevant information within eyespan, the better. Vacant, low-density displays, the posterization of data spread over pages and pages, requires viewers to rely on visual memory—a

weak skill—to make a contrast, a comparison, a choice.” However, if many statements use the Field clause, you may want to describe the Field clause in one place and then refer users to this place.

You can also use sub-diagrams to breakdown a complicated statement. This gives the user a macro view of the statement in the main diagram. For a micro view of the components of the statement, the user can look at the sub-diagrams. Be sure to logically divide the statement according to the components of the statement. Consider how the programmer will be using the statement and use the sub-diagram as a teaching tool. For example, in the WHENEVER statement at the beginning of this paper, you could break out the first list of options, ERROR, WARNING, and NOT FOUND. Or you could break out the other list of options, GOTO, CALL, CONTINUE, and STOP. Using sub-diagrams in this way reinforces the parallelism between the options.

Disadvantages of Using Railroad Diagrams

Railroad diagrams have the following disadvantages:

- Take more space than BNF diagrams.
- Make the replication of the syntax in online documentation difficult.
- Cause users who are familiar with BNF diagrams to learn another syntax diagramming method.

Space

Railroad diagrams require at least 50% more space than the BNF diagrams. For example, in the example shown at the beginning of this paper, the BNF diagram requires 1/2 inch while the railroad diagram requires 1 1/8 inches. Statements with sub-diagrams require even more space.

Online Requirements

Statements throughout a manual set should follow the same syntax conventions. If your manual set includes online documentation, you will need to take special considerations to implement railroad diagrams in an online environment.

If the product you are documenting requires a terminal that supports a graphical user interface (GUI), then you can probably use your original diagrams. However, if you are working in a windowing environment, you need to be sure the diagram fits in the window. If your users can resize the window, you may need to devise a way to wrap the diagram according to the window size. You can also use a hypertext

product and then define pop-up windows for the sub-diagrams.

However, if your users have ASCII terminals, it may be difficult to replicate the railroad diagrams on an ASCII terminal. You could use the hyphen (-), vertical bar (|), and slashes (\and /) to draw the diagrams; however, I’m not sure how usable they would be. If your product requires online documentation, you may want to do some usability studies to ensure the diagrams are usable online in an ASCII environment for your users before converting to railroad diagrams.

Learning Period

If your users are familiar with BNF, there will be a learning curve, as well as an adjustment period. Dave Kirby at Tandem said the programmers opposed to the railroad diagrams in the TACL manual seemed more vehement in their dislike, writing more comments than the programmers who liked the diagrams and writing words in all uppercase, under lining words, and using exclamation marks. Remarks he received include “Don’t insult my intelligence!” and “BNF is familiar.”

Determining Whether to Use Railroad Diagrams

To determine whether your product’s documentation is a good candidate for railroad diagrams, consider the following:

- Your audience
- Your resources
- Your time limitations

Your Audience

What is your audience like? If your audience is sophisticated programmers using many different products that use BNF diagrams in the documentation, you may not want to give them a manual set that is different from the others they regularly use. However, if your programmers use products with railroad diagrams in the documentation, then railroad diagrams may be a good option for you.

To determine if your audience would benefit from railroad diagrams, you may want to do some usability testing with your them. For example, you can watch some typical programmers code a few statements using both BNF and railroad diagrams. Note how they use the two different types

of syntax diagrams. When they finish the statements, ask how they found using the two different types of syntax diagrams. Based on what you saw them do and on what they said they did, you can decide if railroad diagrams are right for your programmers.

Your Resources

What kind of equipment do you have? If you are using a mark-up language such as Scribe or TROFF, including railroad diagrams may be difficult for you. However, if you using a workstation with good graphics software, railroad diagrams should be easier for you to implement. Also, do you have an illustrator that can help you with the diagrams? She or he may be able to help you design and then draw the diagrams.

You also may want to invest some time into developing a tool to help automate the conversion of BNF to railroad diagrams.

Your Time Limitations

Are you on a tight schedule? Designing and then drawing railroad diagrams is a time-consuming process. If you are the first in your company to use railroad diagrams in a manual, you will need to determine the style of the railroad diagrams and then draw diagrams for each statement. Do not forget to include time in the schedule to have the diagrams reviewed and to maintain the diagrams.

Converting BNF to Railroad Diagrams

When converting BNF diagrams to railroad diagrams, be sure to do the following:

- Establish a style guide for the railroad diagrams.
- Make a first draft of each statement track.
- Revise, revise, and revise.
- Be flexible.

Style Guide

If you decide to use railroad diagrams, create a style guide. The guide should determine guidelines such as the following:

- How long each drawing should be.
- What fonts to use for keywords and variables.
- What angle the curves should be.
- How many sub-diagrams a main diagram can include.

If you have an illustrator, be sure to include her or him in the discussion. A style guide is especially important when more than one writer in your company is designing railroad diagrams. By following the style guide, the railroad diagrams will be consistent throughout your company. Tufte says "As our eye moves from one image to the next, this constancy in design allows viewers to focus on changes in information rather than changes in graphical composition. A steady canvas makes for a clearer picture."

If you are using desktop publishing software that includes a template or style option to control the format of a document, create a template for the railroad diagrams. In the template, define the default line width, the size of the arrow heads, and the allowed fonts.

First Draft

After you determine the style guide for the railroad diagrams, start drawing the first drafts of the railroad diagrams. If you are uncomfortable drawing on the computer, you may want to sketch these out on paper first and then transfer them online. When you feel the drawings are accurate, give them to an expert to review. If you have an expert in the publications department, you may want this person to review the diagrams first. As with writing, you want to have as accurate a first draft as possible for your technical reviewers.

Revise

Be aware that you will probably make many revisions to the railroad diagrams. You will probably make substantial revisions after each review. Just as with writing, I was constantly revising the diagram to make the diagram more clear. Try not to make too many aesthetic revisions until the end of the conversion process. All the little changes add up to hours of your time. If a statement is technically wrong, your aesthetic changes are lost.

Be Flexible

Be flexible with your style guide. As you design diagrams for the statements, you may realize that one or two statements require special handling that breaks the style guide. Try only to break the style when necessary and when it makes the diagram more intuitive. You may want to do some usability testing before making any major changes to the style guide. For example, out of the 60 diagrams I created for the 4GL manual, only 2 required special consideration.

Conclusions

Railroad diagrams offer a visual alternative to BNF diagrams. Before deciding to use railroad diagrams, evaluate your product, users, and resources to determine if railroad diagrams are an effective way to depict the statement syntax.

Bibliography

Brockmann, R. John, *Writing Better Computer User Documentation*, John Wiley & Sons, Inc., 1990.

Henno, J., "User Friendly Syntax: Design and Presentation," *International Journal of Man-Machine Studies*, Volume 28, Number 1, January 1988.

Tufte, Edward R., *Envisioning Information*, Graphics Press, 1990.

Thanks to Dave Kirby and Lynn Rollins of Tandem Computers and Jenny Robertson of Informix.