Model Self-Assessment of Writing Principles
Principle 3: Agents, Actions, and Objects

Norman Ramsey

March 2015

Foreword

As always, I was not able to write this model self-assessment as if I were a beginning student. My revision brings into play not only Agents, Actions, and Objects but also Coherent Subjects and Parallel Structure. Regrettably, I find myself unable to escape the influence of these competing principles.

Original and revised texts in context

The text I am using is the second paragraph of a paper I wrote with Carla Marceau in the late 1980s, before I had begun to study any of the methods we are learning in class. The title, abstract, and first paragraph are shown in Figure 1. The original and revised texts of the second paragraph are shown in Figure 2. The paper appeared in the journal *Software—Practice and Experience*. We were aiming at an audience interested in software tools.

Analysis and explanation

Here are the listed elements (a) to (f) of the analysis and explanation called for in the handout.

(a) The agents, actions and objects in my paragraph are shown in Table 3. The important ones are italicized; the others are actions I saw in the original text, but then decided weren't so important.

The triple below the line wasn't in the original paragraph; I pulled it from the abstract of the paper.

(b) It's a bit difficult for me to reconstruct our goals *ex post facto*. The overall goal of the paper was to communicate our experience of what did and did not work well when using literate programming (a software technique supported by tools) on a team project, rather than just as an individual programmer.

This paragraph is one of several intended to help readers figure out what literate programming is, with the hope that readers will decide that literate programming is sufficiently interesting and important that it's worth reading the rest of the paper. The paragraph also tries to lay a little bit of a foundation that will later force the reader to conclude that the experience described in the paper is new.

Analyzing the paragraph in retrospect, I decided that the really important agent is "literate programs"—I want readers to know what literate programs can do. Authors publishing (or not publishing) is also an important agent because I am trying to lay a foundation for saying that the work we did was important even though we had no intent to publish.

The action of using literate programs in teaching (students study programs) is not so important. And at first I thought that Don Knuth is an important agent because any time Don endorses an idea (or, as in this case, originates an idea), the idea has instant credibility. But in the end, I decided what was really important was the distinction between publishing and not publishing.

(c) Subjects and verbs from the original text:

O1. Literate programming is discussed

O2. TEX and METAFONT have been published

O3. Other programs are

O4. Programs were written

O5. Another program is

O6. Another [program] illustrates

(d) Diagnoses:

- The actions "to discuss" and "to write" in sentences O1 and O4 are not well

Literate Programming on a Team Project

Norman Ramsey and Carla Marceau

Abstract

We used literate programming on a team project to write a 33,000-line program for the Synthesizer Generator. The program, Penelope, was written using WEB, a tool designed for writing literate programs. Unlike other WEB programs, many of which have been written by WEB's developer or by individuals, Penelope was not intended to be published. We used WEB in the hope that both our team and its final product would benefit from the advantages often attributed to literate programming. The WEB source served as good internal documentation throughout development and maintenance, and it continues to document Penelope's design and implementation. Our experience also uncovered a number of problems with WEB.

Donald Knuth coined the term "literate programming" when describing WEB, the tool he used to build TEX.[1] He believes that "the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature." Knuth and others have presented examples of such programs.[2–5]
(The sample paragraph appears here)

Figure 1: Context of the text

Original paragraph

O1 Literate programming is usually discussed in the
context of publishing programs or of publish-
O2 ing books or articles about programs. TEX and
METAFONT, the original applications of WEB,
O3 have been published as books.[6,7] Other published
literate programs seem to be primarily for teach-
O4 ing. Programs to find random sequences and
count common words were written to illustrate
O5 the power of literate programming.[2,3] Another
program to count common words is a tutorial on
O6 how to develop and tune a small program.[4] An-
other illustrates formal methods of program de-
velopment and the use of abstract data types.[5]

Revised paragraph

R1 Some literate programs serve primarily as works
R2 of literature—often didactic literature. Such pro-
grams have illustrated the development and tun-
ing of small programs,[4] formal methods of pro-
gram development using abstract data types[5],
and the power of literate programming itself.[2,3]
R3 Other literate programs serve primarily as works
R4 of software. Prime among these are TEX and
METAFONT, which were the original applica-
R5 tions of WEB. All the programs mentioned
R6 above have been published. But even when
not published, a literate program can provide
valuable documentation throughout development
and maintenance—as our experience shows.

Figure 2: Original and revised texts

Agent	Action	Object
Authors	*publish*	*literate programs*
Authors	*don't publish*	*literate programs*
Knuth	published	TEX and METAFONT
Students	study	literate programs
Literate programs	*illustrate*	*technique*
Literate programs	*illustrate*	*algorithms*
Literate programs	*illustrate*	*literate programming*
Literate programs	*solve*	*real problems*
Literate programs	*run*	
People	*read*	*literate programs*
Literate programs	*document*	*themselves*

Figure 3: Agents, actions, and objects (important ones italicized)

aligned with the agents and actions in my table. "Discuss" in particular is far off the mark.

- Sentences O3 and O5 have no actions; in both sentences, the main verb is "is."

The subjects and verbs in sentences O2 and O6 are OK. (N.B. In sentence O2, the object is the grammatical subject, not the agent. I was trying to make subjects coherent.)

(e) Here are the grammatical subjects and verbs in the revised text:

R1. Programs serve

R2. Programs have illustrated

R3. Programs serve

R4. TEX and METAFONT are

R5. Programs have been published

R6. A literate program provides

(f) Only sentences R2 and R6 respect the principle.

Sentence R5 comes close to respecting the principle, but in order to improve information flow, it's been placed into the passive voice, with an object acted upon as the grammatical subject.

Self-review

Problems diagnosed in the revised text:

- Although sentences R1 and R3 show parallel structure, sentences R2 and R4 do not—and I wish they did.
- I'm not certain that anything in R4 corresponds to an important action.

Level of mastery:

- Many parts of the text respect the principle, but for reasons that seem strong to me, there are some parts of the text in which I have chosen to disregard the principle.

  Sentences R2 and R6 respect the principle. It's worth noting that sentence R2 condenses actions from original sentences O4, O5, and O6.

  I have chosen to disregard the principle in sentences R1, R3, and R5. And in sentence R4, I was not able to find a way to respect the principle.

  I chose to disregard the principle in order to prioritize coherent subjects and parallel structure. "Literate programs" are the subject of sentences R1, R3, and R5.

  Sentence R1 expresses the action "people read literate programs," and sentence R3 addresses the actions "literate programs solve problems" and "literate programs run." But the actual verb used is the weak verb "serve as." In exchange for this weak verb, what I get is the ability to use "literate programs" as the subject of each sentence, plus the parallel structure. I like the tradeoff.

Summary judgement:

- The text says mostly what I want it to say, and I can't get it any better.

  I feel that I've shown a solid understanding of the principle, but in the majority of the

sentences, I've either been unable to apply it or I've chosen not to apply it. For the work presented here, I would give myself an assessment of "partial mastery": everything is pretty good except for that pesky R4.

Results of peer review

Coming.

Principle 3: Agents, Actions, and Objects

Material for peer review

Norman Ramsey

Context

The text I am using is the second paragraph of a paper I wrote with Carla Marceau in the late 1980s, before I had begun to study any of the methods we are learning in class. The paper appeared in the journal *Software—Practice and Experience*. We were aiming at an audience interested in software tools. The title, abstract, and first paragraph are shown in Figure 4 on Page 6.

Original paragraph

O1 Literate programming is usually discussed in the context of publishing programs or of publishing
O2 books or articles about programs. TEX and METAFONT, the original applications of WEB, have been
O3 published as books.[6,7] Other published literate
O4 programs seem to be primarily for teaching. Programs to find random sequences and count common words were written to illustrate the power
O5 of literate programming.[2,3] Another program to count common words is a tutorial on how to de-
O6 velop and tune a small program.[4] Another illustrates formal methods of program development and the use of abstract data types.[5]

Revised paragraph

R1 Some literate programs serve primarily as works
R2 of literature—often didactic literature. Such programs have illustrated the development and tuning of small programs,[4] formal methods of program development using abstract data types[5], and the
R3 power of literate programming itself.[2,3] Other literate programs serve primarily as works of software.
R4 Prime among these are TEX and METAFONT,
R5 which were the original applications of WEB. All the programs mentioned above have been pub-
R6 lished. But even when not published, a literate program can provide valuable documentation throughout development and maintenance—as our experience shows.

Mastery statement and summary judgement

Level of mastery:

- Many parts of the text respect the principle, but for reasons that seem strong to me, there are some parts of the text in which I have chosen to disregard the principle.

  Sentences R2 and R6 respect the principle. It's worth noting that sentence R2 condenses actions from original sentences O4, O5, and O6.

  I have chosen to disregard the principle in sentences R1, R3, and R5. And in sentence R4, I was not able to find a way to respect the principle.

  I chose to disregard the principle in order to prioritize coherent subjects and parallel structure. "Literate programs" are the subject of sentences R1, R3, and R5.

  Sentence R1 expresses the action "people read literate programs," and sentence R3 addresses the actions "literate programs solve problems" and "literate programs run." But the actual verb used is the weak verb "serve as." In exchange for this weak verb, what I get is the ability to use "literate programs" as the subject of each sentence, plus the parallel structure. I like the tradeoff.

Summary judgement:

- The text says mostly what I want it to say, and I can't get it any better.

  I feel that I've shown a solid understanding of the principle, but in the majority of the sentences, I've either been unable to apply it or I've chosen not to apply it. For the work presented here, I would give myself an assessment of "partial mastery": everything is pretty good except for that pesky R4.

Literate Programming on a Team Project

Norman Ramsey and Carla Marceau

Abstract

We used literate programming on a team project to write a 33,000-line program for the Synthesizer Generator. The program, Penelope, was written using WEB, a tool designed for writing literate programs. Unlike other WEB programs, many of which have been written by WEB's developer or by individuals, Penelope was not intended to be published. We used WEB in the hope that both our team and its final product would benefit from the advantages often attributed to literate programming. The WEB source served as good internal documentation throughout development and maintenance, and it continues to document Penelope's design and implementation. Our experience also uncovered a number of problems with WEB.

Donald Knuth coined the term "literate programming" when describing WEB, the tool he used to build TEX.[1] He believes that "the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature." Knuth and others have presented examples of such programs.[2–5]
(The sample paragraph appears here)

Figure 4: Context of the text