Writing with Style

William Cook
University of Texas at Austin

based on
Style: Toward Clarity and Grace
by Joseph Williams
Encode
a complex web of ideas...
...as a linear stream of text
paper organization ≠ research process
<table>
<thead>
<tr>
<th>Significance</th>
<th>Motivate why the research is important or useful. <strong>Explain</strong> what problem it addresses.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarity</td>
<td>Organize the paper well and write clearly. Make sure you support your claims.</td>
</tr>
<tr>
<td>Novelty</td>
<td>Extend the frontier of knowledge. Explicitly relate your research to <strong>previous work</strong>.</td>
</tr>
<tr>
<td>Correctness</td>
<td>Critically evaluate and support your claims with proofs, an implementation, examples, or experiments.</td>
</tr>
</tbody>
</table>
Clarity
• Subject of sentence names a character

• Verbs name action involving characters
Missing Subjects

“Termination occurred after 23 iterations”

“The program terminated after 23 iterations”
Missing Subjects

“Determination of policy occurs at the presidential level”
Subject=Actor

“The President determines policy”
Weak Verbs

“The algorithm supports effective garbage collection in distributed systems”
Stronger

“The algorithm collects garbage effectively in distributed systems”
NOM:
Nominalization

Noun instead of verb/adjective
## Verb NOM

<table>
<thead>
<tr>
<th>Verb</th>
<th>Nominalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>discover</td>
<td>discovery</td>
</tr>
<tr>
<td>move</td>
<td>movement</td>
</tr>
<tr>
<td>collaborate</td>
<td>collaboration</td>
</tr>
</tbody>
</table>
## Adjective NOM

<table>
<thead>
<tr>
<th>Adjective</th>
<th>Nominalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>difficult</td>
<td>difficulty</td>
</tr>
<tr>
<td>applicable</td>
<td>applicability</td>
</tr>
<tr>
<td>different</td>
<td>difference</td>
</tr>
</tbody>
</table>
empty verb + NOM

“The police conducted an investigation of the matter”
Verb=Action

“The police investigated the matter”
“there is” + NOM

“There is a need for further study of this program”
Name the Actor

“The engineering staff must study this program further”
NOM + empty verb

“The intention of the IRS is to audit our records”
Verb=action

“The IRS intends to audit our records”
NOM + NOM

“There was a review of the evolution of the technique”
Find Actor

“She reviewed the evolution of the technique”
Using “how”

“She reviewed how the technique evolved”
NOM + verb + NOM

“Extensive rust damage to the hull prevented repairs to the ship”
“Because rust had damaged the hull, we could not repair the ship”
“object representation”

(how the object is represented)
Useful Nominalizations
Reference to previous sentence

“these arguments all depend upon…”

“This decision has…”
Name a verb’s object

“I do not understand her meaning or his intention”

(what she means what he intends)
Common concepts

“Taxation without representation was not the central cause of the revolution”
compilation
dependency
inheritance
implementation
Cohesion

Managing the flow of information
Sentences

- ideas already mentioned
- familiar ideas

- action
- new ideas
Topics form a logical sequence of ideas
Technique

Underline subjects

Do they match?
The quality of error reporting of modern software has decreased as its complexity has increased. End-users take the cryptic error messages given to them by programs and struggle to fix their problems using search engines and support websites. Developers cannot improve their error messages when they receive an ambiguous or otherwise insufficient error indicator from a black-box software component.
An error occurs when software cannot complete a requested action as a result of some problem with its input, configuration, or environment. A high-quality error report allows a user to understand and correct the problem. But the quality of error reports has been decreasing as software becomes more complex and layered. End-users take the cryptic error messages given to them by programs and struggle to fix their problems using search engines and support websites. Developers cannot improve their error messages when they receive an ambiguous or otherwise insufficient error indicator from a black-box software component.
Emphasis

Put important things at the end

<table>
<thead>
<tr>
<th>sentence</th>
<th>final words</th>
</tr>
</thead>
<tbody>
<tr>
<td>paragraph</td>
<td>last sent.</td>
</tr>
<tr>
<td>section</td>
<td>last para.</td>
</tr>
</tbody>
</table>
Coherence
The Point

Intro  | Discussion

---|---
The point (best) | ...or here (ok)
Containers
• Large-scale Structure
• Sequence of items

Specific rules

Paper
Section
Paragraph
Sentence
Themes

Strings of related words
Woven into the text
Active

Passive
Active

<table>
<thead>
<tr>
<th>subject</th>
<th>The partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb</td>
<td>broke</td>
</tr>
<tr>
<td>object</td>
<td>the agreement</td>
</tr>
</tbody>
</table>
Passive

Subject: The agreement

Verb: was broken

Prepositional Phrase: the partners
Passive is fine, if it is more coherent.
“Our partners were old friends... but they let us down. The partners broke the agreement.”
Passive

“We thought we had a good agreement. Then we found out who killed it. The agreement was broken by the partners.”
Miscellaneous Rules
Section Title Rule

First sentence of every section: Must include the section title

(except intro/conclusion)
Little Piggy Rule

Avoid “we” as subject, unless it is something you the author actually did.

On the other hand, in mathematical writing, “we” is accepted usage.

(This little piggy went we... we.. we... all the way home)
“Our” Rule

Avoid “our”, as in “our technique”

Give everything a name instead
“This” Rule

Avoid “this” as a subject.

Or qualify it: “this technique”..
Summary
Connect your sentences

Define your terms

Expose hidden assumptions

Get a second reader

Learn how to recognize problems
Exercises

(apologies to Doug and Emmett)
The PowerPC ISA has some features that make it different from the Alpha and PISA ISAs. For example, the Alpha ISA has 25 instructions with 4 formats and the PISA ISA has 135 instructions with 4 formats. Not all of these instructions are implemented in the simulator. In this section, we describe features of the ISA that are implemented in the simulator.
3 TRIPS Architecture

The TRIPS architecture is designed to address key challenges posed by next-generation technologies—power efficiency, high concurrency on a latency-dominated physical substrate, and adaptability to the demands of diverse applications [10, 12]. It uses an EDGE ISA [2], which has two defining characteristics: *block atomic execution* and *direct instruction communication*. The ISA aggregates large groups of instructions into blocks which are logically fetched, executed, and committed as an atomic unit by the hardware. This model amortizes the cost of per-instruction overheads such branch predictions over a large number of instructions. With direct instruction communication, instructions within a block send their results directly to the consumers without writing the value to the register file, enabling lightweight intra-block dataflow execution.
4 Exit Predictor Design

In this section we describe hyperblock-based exit predictors in detail. We design exit predictors based on both conventional schemes and neural techniques. Exit predictors based on conventional techniques have a simple and scalable design, and can make fast predictions, with accuracies close to some of the best traditional branch predictors. The perceptron-based exit predictor requires more time to make a single prediction, but provides higher accuracy than other high-bandwidth exit predictors.
3 Critical Path Model

The critical path model for the TRIPS architecture is heavily based on the dependence-graph model previously developed for superscalar architectures [5]. The model represents various microarchitectural events as nodes in a directed acyclic graph. Edges between the nodes represent dependence constraints among the events. Figure 2 shows a typical dependence graph constructed for a slice of four blocks seen during the program execution. In addition to representing the usual constraints such as data dependences, branch mispredictions, and finite instruction window sizes, the TRIPS model represents constraints imposed by block atomic execution and operand routing.