The Data Over Code Principle: Readings

On the Criteria to be Used in Decomposing Systems Into Modules.

The decomposition still contains some globally shared assumptions, which may or may not be reasonable. They might or might not impact the interoperability. For example:

- What is a word? In some languages it is not obvious where words start and end.
- What is a letter? Some languages have combinations of letters (digraphs or trigraphs, or poorly encoded ligatures) that must be kept together and even sorted in a different position. Some languages consider different letters to just be graphical variations of the same letter.
- Characters are returned as integers, or is it bytes? What is the encoding? Some encodings also have variable length.

Another possible problem could be modification of state, especially combined with assumptions about which order things are called in.

The Secret History of Information Hiding.

Abstraction: Not What You Think It Is.

See examples

Research Corner: The Programmer's Apprentice.

The dataflow graph for creating the sum can be isolated and identified from other code that is not relevant for creating the sum.

A very practical way to do this is to trace the graph backwards. Start at the output node ”sum”, colour it red. Move back along the incoming edges to the next set of nodes and colour them red also. Continue until you have reached all nodes that you can in this way. That subgraph induced on the red nodes is the ”sum” function. Delete all the variables that correspond to uncoloured edges.

###Extra info for the curious: This is a version of slicing. Specifically, it is a thin slice