software studio

separation of concerns

Daniel Jackson

where the term came from

Let me try to explain to you, what to my taste is characteristic for all intelligent thinking. It is, that one is willing to study in depth an aspect of one's subject matter in isolation for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects... It is what I sometimes have called "the **separation of concerns**", which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of.

—Edsger Dijkstra; On the role of scientific thought; EWD447; 30th August 1974

SoC in behavior design

helps you focus

› easier to pay attention to one thing at a time

› put some complexities aside

encourages decoupling

› disentangle aspects that seemed intertwined

› can consider cross product of feature options

example: arguments in HTTP requests

verb	*where*	*property*	*visibility*
GET	query string	no side effects	visible
POST	request body	side effects	invisible

SoC in architectural design

separate services by function

› eg: searching, ordering, billing, advertising

separate out critical functions

› eg: "never deliver more than maximum dose"

separate features

› eg: presentation, behavior, data

SoC in code design

separate spec from implementation
› eg: data semantics from data representation
› eg: post-condition from algorithm

what does this spec say?
what does it *not* say?

keys

```
public abstract Enumeration<K> keys()
```
Returns an enumeration of the keys in this dictionary. The general contract for the keys method is that an `Enumeration` object is returned that will generate all the keys for which this dictionary contains entries.

Returns:

an enumeration of the keys in this dictionary.

See Also:

`elements()`, `Enumeration`

6.170 Software Studio

Spring 2013