Separation of Concerns

Hao Zhe Kok, Binbin Wu, Ramsey Towell

What?

Separation of concerns allows us to deal with different aspects of a problem, so that we can concentrate on each individually.

Separation of concerns is a commonsense practice that we try to follow in our everyday life to overcome the difficulties we encounter. The principle should be applied also in software development.

Manifested in modularity and encapsulation.

What?

The most important application of separation of concerns is to separate problem-domain concerns from implementation-domain concerns.

Problem-domain properties hold in general, regardless of the implementation environment.

Separation of concerns may result in separation of responsibilities in dealing with separate issues which results in dividing the work on a complex problem into specific assignments, possibly for different people with different skills.

MVC

How?

There are many decisions that must be made in the development of a software product.

Some of them concern features of the product: functions to offer, expected reliability, efficiency with respect to space and time, user interfaces, etc.

Some people are concerned with the development process: the development environment, the organization and structure of the teams, scheduling, control procedures, design strategies, error recovery mechanisms, etc.

Others concern economic and financial matters.

How?

The only way to master the complexity of a project is to separate the different concerns. First of all, we should try to isolate issues that are not so closely related to the others. Then, we consider issues separately, together with only one relevant details of related issues.

First type: One can separate them in Time.
Second type: One can be separated in terms of qualities.
Third type: Separation of concerns allows different views of the software to be analyzed separately.
Fourth type: Separation of concerns allows us to deal with parts of the same system separately; here, separation is in terms of size.

Why not?

There is an inherent disadvantage in separation of concerns: By separating two or more issues, we might miss some global optimization that would be possible by tackling them together.

But our ability to make “optimized” decisions in the face of complexity is rather limited. If we consider too many concerns simultaneously, we are likely to be overwhelmed by the amount of detail and complexity we face.

System designers and architects often face such trade-offs.