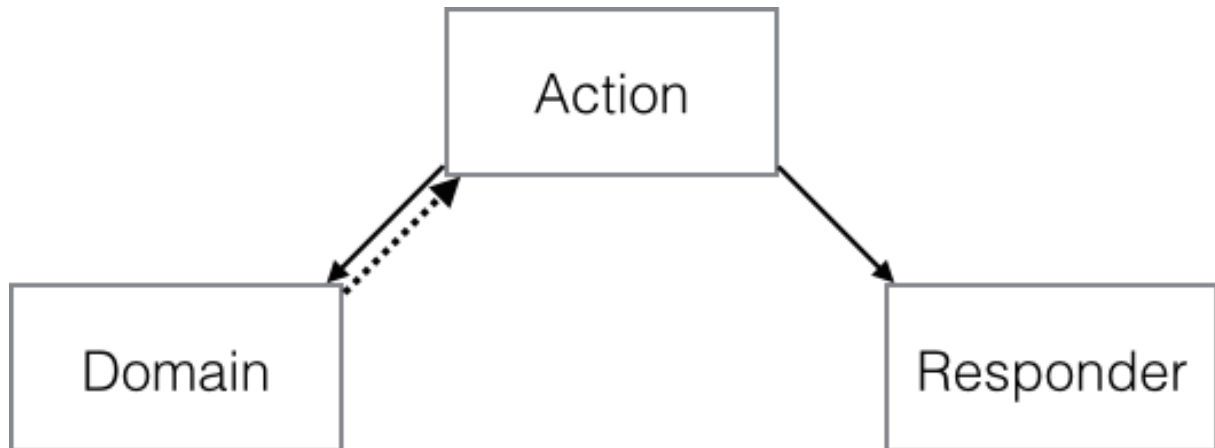

Action Domain Responder

Action Domain Responder organizes a single user interface interaction between an HTTP client and a HTTP server-side application into three distinct roles.



Components

Action is the logic to connect the *Domain* and *Responder*. It invokes the *Domain* with inputs collected from the HTTP Request, then invokes the *Responder* with the data needed to build an HTTP Response.

Domain is an entry point to the domain logic forming the core of the application. It may be a *Transaction Script*, *Service Layer*, *Application Service*, or something similar.

Responder is the presentation logic to build an HTTP Response using data it receives from the *Action*. It deals with status codes, headers and cookies, content, formatting and transformation, templates and views, and so on.

Collaborations

1. The web handler receives an HTTP Request and dispatches it to an *Action*.
2. The *Action* invokes the *Domain*, collecting any required inputs to the *Domain* from the HTTP Request.
3. The *Action* then invokes the *Responder* with the data it needs to build an HTTP Response (typically the HTTP Request and the *Domain* results, if any).
4. The *Responder* builds an HTTP Response using the data fed to it by the *Action*.
5. The *Action* returns the HTTP Response to the web handler sends the HTTP Response.

Reading

- *Model View Controller* and “Model 2”
- Comparing “Model 2” MVC to ADR
- Tradeoffs in ADR
- Objections to ADR

Resources

- Refactoring from “Model 2” MVC to ADR
- Implementation notes and advice
- ADR discussions, mentions, implementations, etc.

Example code resides in the [example](#) repository.

You can also find out about the history of this pattern, the research bibliography, and the author’s acknowledgments.