+ public

- private

# protected

~ package visible

* Controller: Object implementing this role makes decisions and closely directs the action of other objects.
* Coordinator: This role reacts to events by delegating tasks to others.
* Information Holder: Information holder knows and provides information.
  + Information Provider: A slight variation of an information holder is the information provider, which takes a more active role in managing and maintaining information. This distinction can be used if a designer needs to get more specific.
* Interfacer: This role transforms information and requests between distinct parts of an application. It is further divided into more specific roles.
  + External Interfacer: External interfacer communicates with other applications rather than its own. It is mainly used for encapsulating non-object-oriented APIs and does not collaborate a lot.
  + Internal Interfacer: Also called intersystem interfacer.[[14]](https://en.wikipedia.org/wiki/Responsibility-driven_design#cite_note-ObjectDesign-page93-14) It act as a bridge between object neighborhoods.
  + User Interfacer: User interfacer communicates with users by responding to events generated in the UI and then passing them on to more appropriate objects.
* Service Provider: This role performs work and offers computing services.
* Structurer: This role maintains relationships between objects and information about those relationships.