NgRx is a framework for creating reactive web applications with Angular. It accomplishes this by implementing the Flux-Pattern which introduces a global/common state in an Angular application. This allows for better use of data over multiple components.

Reactive applications can be defined from The Reactive Manifesto as applications that are responsive, resilient, elastic, and message driven. In a reactive Angular application, chances are that data will need to be shared between many different components. By implementing a global state that can hold active data, different components can each access the same data easily after it has been read from an origin such as a database once. A good example of this in practice is an ecommerce site adding an item to a user’s shopping cart. The data about the item being added has already been read when it was called to be shown to the user. Further, there may have been changes when the user added it to the cart such as a choice of color or clothing size. By implementing a global state with NgRx, the necessary data is already available and being stored in the global state with other common data. This is accomplished with the Flux-Pattern.

The Flux Pattern implemented by NgRx is based on the principle of having data flow in a single direction. The components of this are actions, the store, reducers, selectors, and effects. Simply put, the store is the central component of NgRx that provides a single global state for all components of an application to easily access. Other components such as actions and selectors are self-explanatory at least at a basic level. Reducers handle state transitions from the store which are in the form of functions, and effects use streams to provide sources for actions which isolate side effects from components.

References

Roos, P. (2022, September 17). *What is NgRx and why is it used in Angular apps?* workingsoftware.dev. https://www.workingsoftware.dev/what-is-ngrx-and-why-is-it-used-in-angular/