

Chain

The Gang of Four's Chain of Responsibility (COR) behavioral design pattern is interesting and commonly used to model a set of operations. Each operation needs to be modeled as a command object and linked to others to form a chain. The command objects act upon some data and indicate whether the control should get passed to the next command in the chain. The benefit is that an operation can be added or removed easily without changing code. Jakarta Commons Chain models this design pattern as a reusable library.

You only need to provide individual objects which implement the Command interface and which make use of a Context object passed to each command in order to access shared data between commands. You can use a XML file to describe sequences of commands (i.e. a Chain) and assign them a name in order to invoke them easily or use them as individual commands within other chains.

Note: Chain is not a full workflow system. It does not have sophisticated flow of control between branches of chains or looping. However, it is also easier to get started with than most workflow systems and you can easily refactor existing code to fit within it. Editing the XML file to change the sequence of commands or alter the parameters passed to a command is also well within the realm of what many end users can do themselves without minimal instruction.

A Devx article with a simple one page overview of Chain is available here:

- <http://www.devx.com/Java/Article/29392/0/page/4>

A fairly comprehensive pair of articles which focus only on the Commons Chain are at O'Reilly's [OnJava](#) site:

- [ONJava.com – A Look at Commons Chain, Part 1](#)
- [ONJava.com – A Look at Commons Chain, Part 2](#)