WARPLANE FACTORY

Statement of Work:

In this project, there are different warplanes seperated into models (F-16, F-35 etc.) and types (Fighter, Bomber etc.).Planes are being created factories. Factories may have some common processes but each factory has it's own creation. Each different plane has it's own weapons. For example, if the product is an F-16 Bomber, it's weapons will be for bombing and suitable for an F-16. In addition to this, each plane can fly in different styles (supersonic, subsonic, chase mode etc.). Their flying behaviors should also be interchangeable and can be changed whenever needed. For example a specific plane can fly supersonic or subsonic according to the situation.

Design Patterns:

Factory method and strategy patterns are useful to solve this problem.

Factory method pattern: We can let subclasses decide how to create their own products. We can define an abstract create method to a factory supertype, and it's subclasses can create their own plane models with their types. We can also program common processes to the supertype.

Strategy pattern: Since each plane can have different and interchangeable flying behaviors, we can encapsulate those behaviors. We can create flying behaviors seperately and make them implement an interface. Also a warplane should "has" a flying behavior. So all warplanes can "fly" but they also know their flying behavior.

UML: