



What are patterns – Part III?

Patterns as a Roadmap
in complex designs

*Why do we then need design patterns
when most of them are simply the
application of three fundamental
principles?*

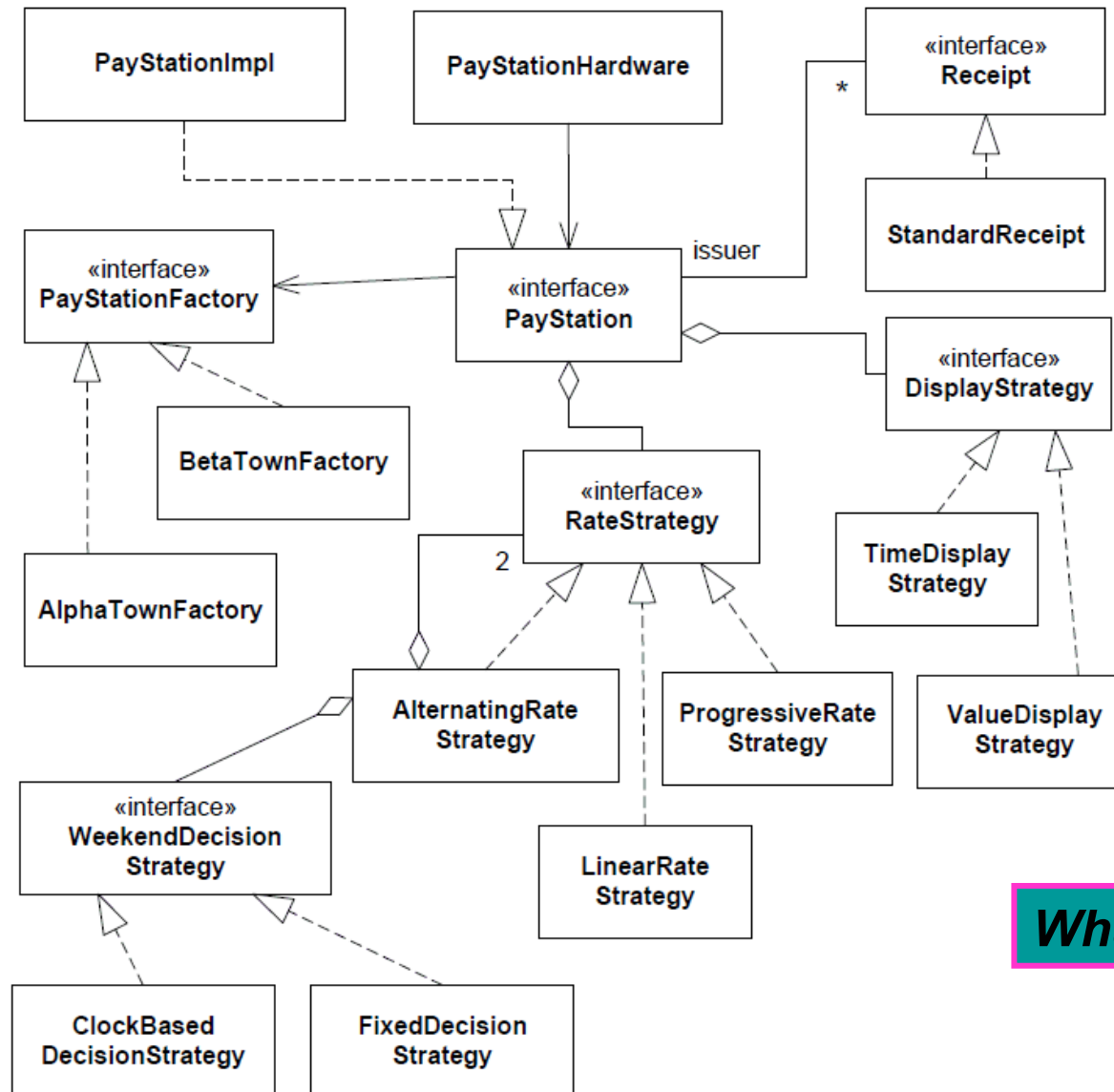




For defining a terminology of **why** we have used the principles in this way!

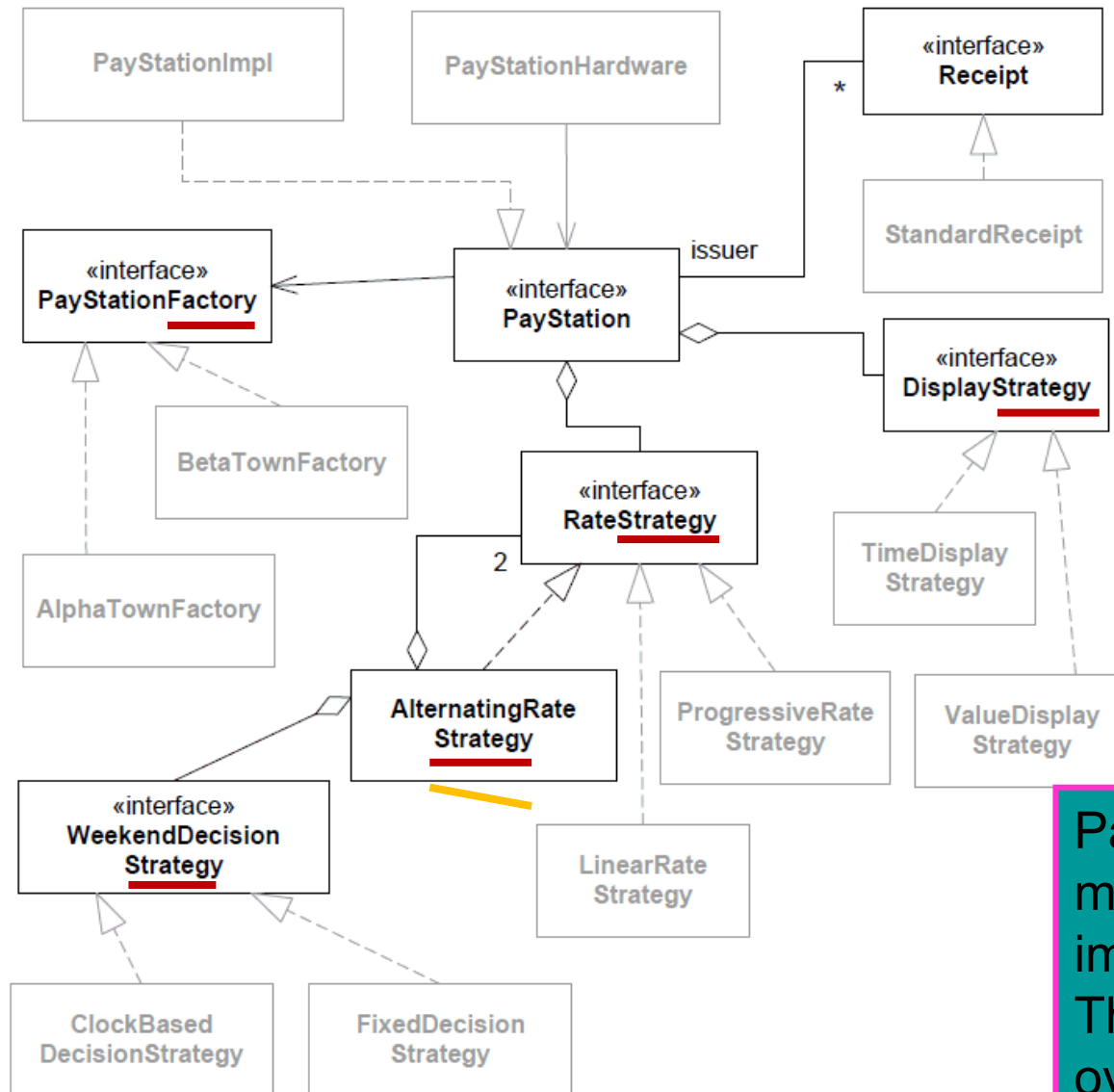
- for documentation purposes
- for use during design discussions
- for finding our way in a big mess of classes !

Example



What a mess !!!

Patterns for sense-making



Patterns define the road map for all the implementation classes. Thus we maintain overview!

Yet another definition

Definition: Design Pattern (Roadmap view)

Design patterns structure, document, and provide overview of the roles and protocols in complex, compositional, designs. A design pattern serves as a roadmap of a part of the design.

Basically patterns **classify and document** how the principles have been applied – they provide the **link between the solution and the problem**

Thus it is important – in my view – to include the pattern name in the interface/class names

... and describe the roles in the API documentation (JavaDoc, etc.)

Beware of programmers that are **not trained** in patterns. They **do** see a big mess! And make one!!!

... but



Of course this is a bit of an over-generalisation.

Some patterns are more complex and have more explicit behavioural constraints

- Visitor, Observer, Mediator, etc.

However, you will see the principles in action even in these patterns...