Software Engineering Essentials

Patterns

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics

Learning goals

1) Understand why patterns are useful in software engineering
2) Analyze the difference between a pattern and an algorithm
3) Remember the categorization of patterns

Algorithm vs. pattern

- **Algorithm:**
  - A method for solving a problem using a finite sequence of well-defined instructions for solving a problem
  - Starting form an initial state, the algorithm proceeds through a series of successive states, eventually terminating a final state
- **Pattern:**
  - *„A pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem in such a way that you can use this solution a million times over, without ever doing it the same way twice“* - Christopher Alexander, A pattern language.

Pattern definition

Original definition (Christopher Alexander):

A pattern is a three-part rule, which expresses a relation between a certain ***context****, a* ***problem****, and a* ***solution****.*

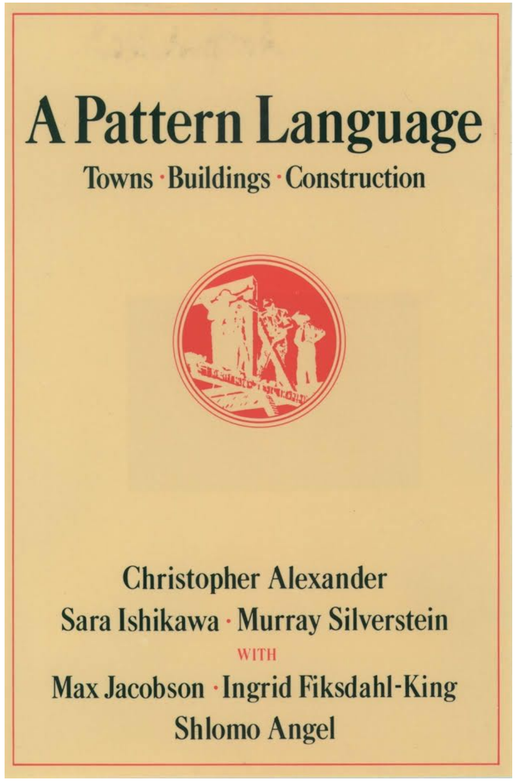
Patterns originated in architecture

Christopher Alexander

* 1936 Vienna, Austria

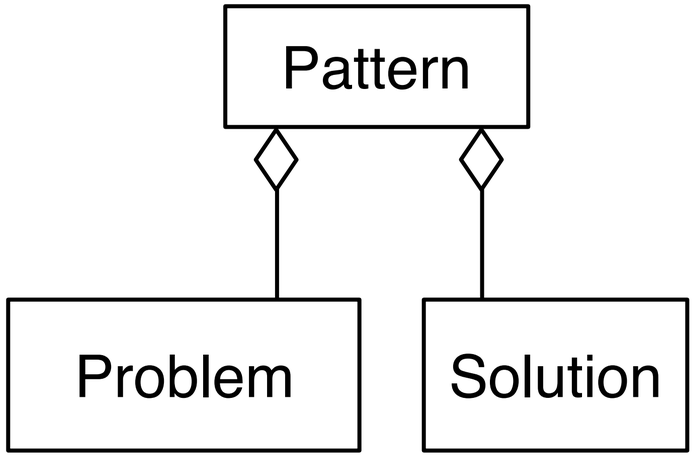
Over 200 building projects

Author of the book “A Pattern Language”



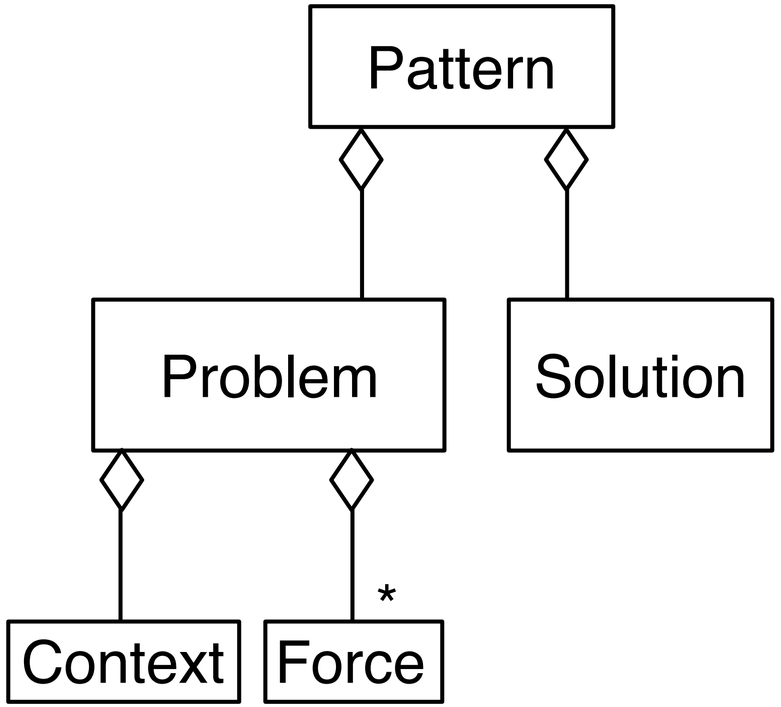
Pattern definition 1/5

- A pattern has a problem and a solution



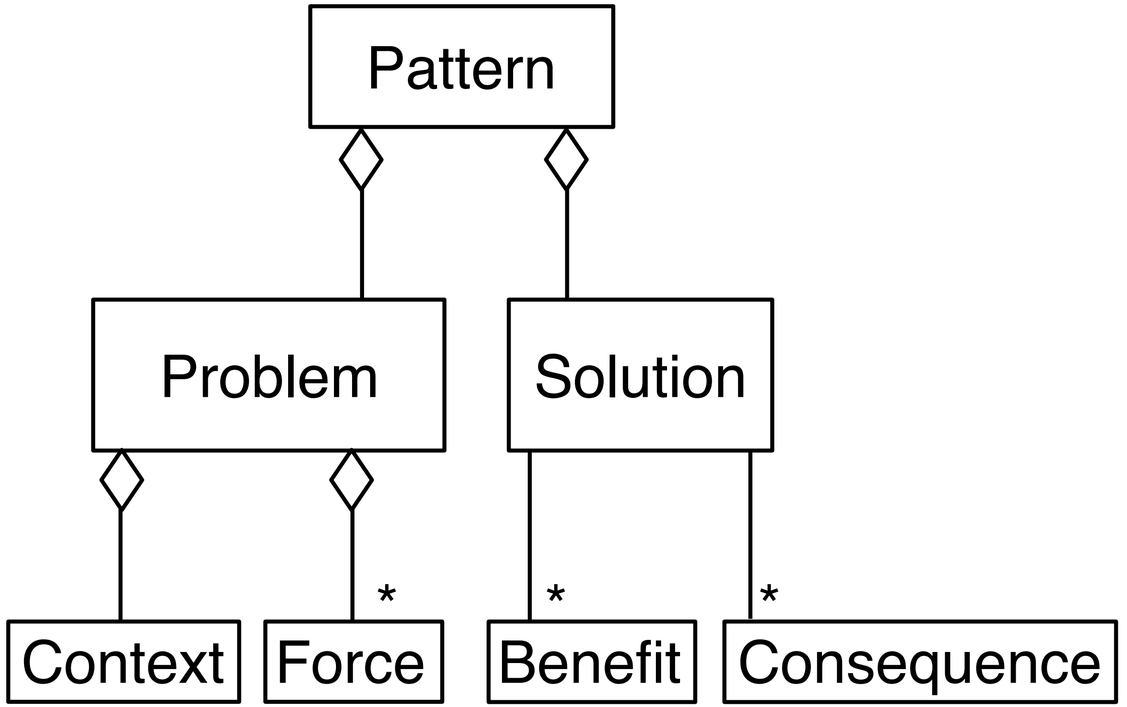
Pattern definition 2/5

- A pattern has a problem and a solution
  - The problem class is elaborated in terms of a context and a set of forces



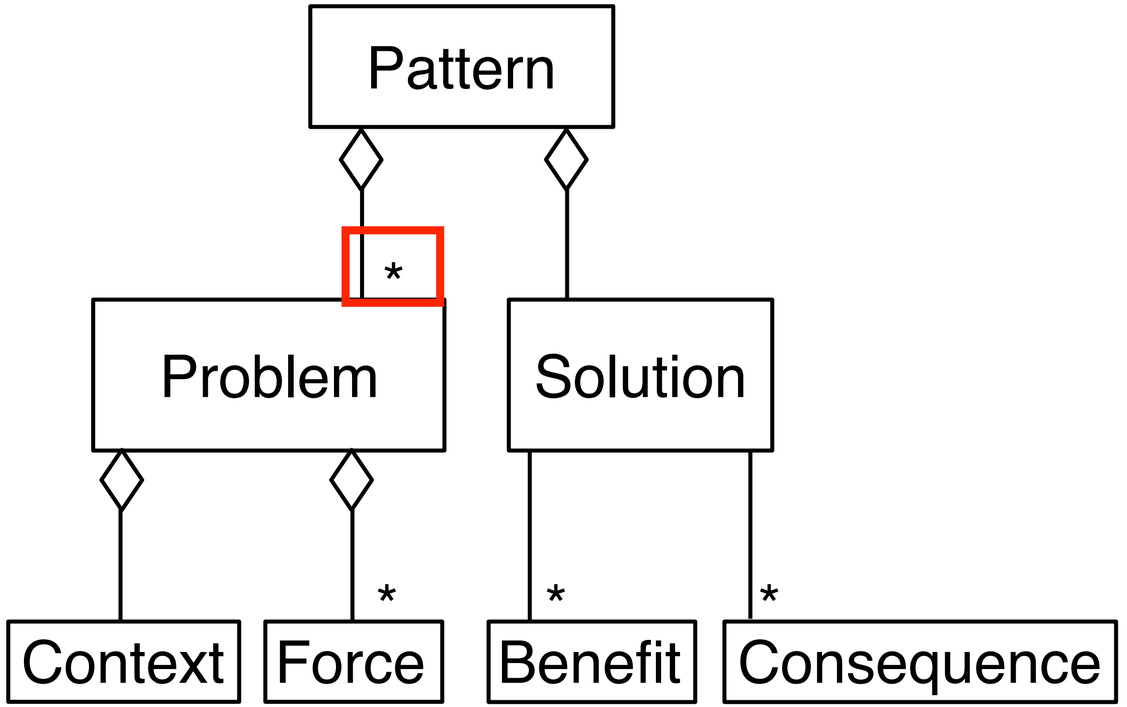
Pattern definition 3/5

- A pattern has a problem and a solution
  - The problem class is elaborated in terms of a context and a set of forces
  - The solution resolves these forces with benefits and consequences



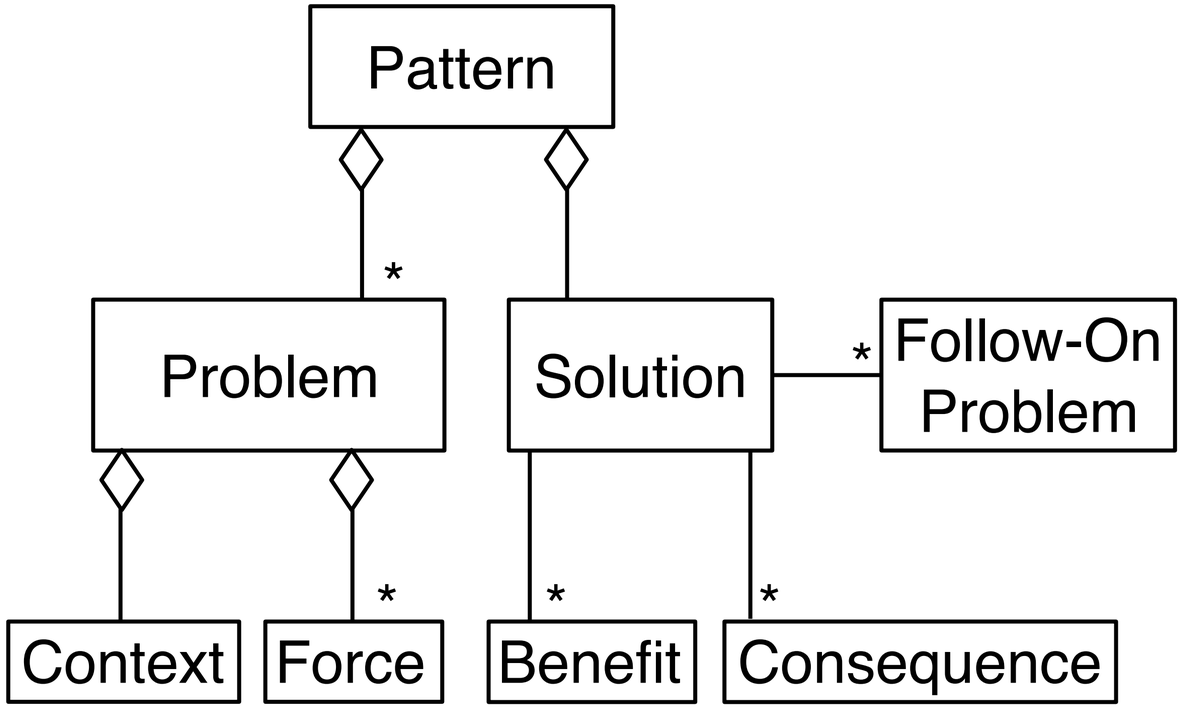
Pattern definition 4/5

- A pattern has a problem and a solution
  - The problem class is elaborated in terms of a context and a set of forces
  - The solution resolves these forces with benefits and consequences
  - To be considered as a pattern, the solution must be applicable to more than one specific problem



Pattern definition 5/5

- A pattern has a problem and a solution
  - The problem class is elaborated in terms of a context and a set of forces
  - The solution resolves these forces with benefits and consequences
  - To be considered as a pattern, the solution must be applicable to more than one specific problem
- Solutions usually generate follow-on problems
  - Follow-on problems can again be elaborated in terms of context and forces which may lead to the applicability of other patterns

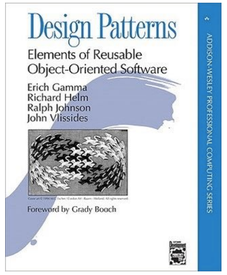


3 schemata for describing patterns

1) Alexandrian form (architecture)

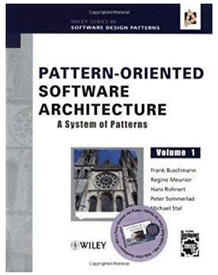
2) Gang of four: name, intent, also known as, motivation, applicability, …

Erich Gamma et. al



3) Gang of five: name, also known as, example, context, problem, …

Frank Buschmann et. al



Categorization used for this course

- **Patterns for development activities**
  - Analysis
  - Architecture
  - Design
  - Testing
- **Patterns for cross-functional activities**
  - Process
  - Agile
  - Build and release management
- **Antipatterns**
  - Smells & refactorings

Software Engineering Essentials

Patterns

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics