Model - the UML model of the tested application. 

The system will be compared with older models. Simultaneously will provide information about common problems and other implemented patterns.

The context in which each problem resides will be compared with known contexts for patterns to see what patterns could solve it.

The relevant parts of the model will be stored.

After applying the solution steps have to be repeated since a known context for patterns to see what pattern could solve it.

To be able to maintain the functionality of the processed model the "compiler" will apply patterns using strategies that are a combination of refactorings and aspect-oriented constructs.

The abstraction level is also associated with a different role in the system of patterns. Patterns that belong to the architectural layer will divide a model into subsystems and components.

Another basic category is the Pattern. This is the type for any pattern and antipattern used in the system. The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The system of patterns must be specialised on a particular technology in order to improve the performance of software.

The goal is to build a system of patterns that can be used to automatically improve the performance of software.

The solution is proposed as a sequence of refactorings. The block of dead code is extracted into a separate method as defined by Extract Method refactoring. Finally the method is moved into another class and removed.

The context in which each problem relies will be compared with patterns and antipatterns.

The system of patterns must be built with patterns that can correct this kind of performance problems. Other patterns must be included to coagulate the system.

The relevant parts of the model will be stored.