LIBRARY
Michigan State
University

This is to certify that the
thesis entitled

USING EVOLUTIONARY COMPUTATION TO AUTOMATICALLY REFACTOR SOFTWARE DESIGNS TO INCLUDE DESIGN PATTERNS

presented by

ADAM C. JENSEN

has been accepted towards fulfillment
of the requirements for the

MASTER OF SCIENCE degree in COMPUTER SCIENCE

Betty H C Cheng
Major Professor's Signature

4/15/10
Date

MSU is an Affirmative Action/Equal Opportunity Employer

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

5/08 K:/Proj/Acc&Pres/CIRC/DateDue.indd

Using Evolutionary Computation to Automatically Refactor Software Designs to Include Design Patterns

By

Adam C. Jensen

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Computer Science

2010

ABSTRACT

Using Evolutionary Computation to Automatically Refactor Software Designs to Include Design Patterns

By

Adam C. Jensen

Building a coherent object-oriented software design is a difficult, time-consuming task. Designing maintainable and extensible software that can be revised to satisfy new requirements is even more difficult. Recent approaches view software design as an optimization problem, thus enabling techniques such as evolutionary computation for automating design improvement. These approaches focus on making small, behavior-preserving changes to software designs, such as moving an operation from one class to another and updating references to that method. However, these approaches do not leverage larger design strategies, such as design patterns. This thesis introduces an approach for automated improvement of software designs that focuses on the introduction of design patterns through refactoring. The proposed approach applies genetic programming, an evolutionary computation technique, to identify suitable refactorings that improve the quality of a software design with respect to a set of software engineering metrics. By automatically introducing design patterns and using object-oriented metrics to guide the evolutionary process, our approach is able to generate a broader set of refactoring solutions that incorporate design patterns than traditional refactoring techniques. We apply this approach to a large set of published software models, as well as a model derived from a Web-based application.

ACKNOWLEDGEMENTS

To acknowledge everyone who contributed to the work in this thesis, I would need to mention everyone who interacted with me – directly or indirectly, deliberately or accidentally – in the past fifteen months. I thank each and every one of you. In any creative endeavor, the novelty and randomness of our surrounding environment plays a crucial role in motivating us, drawing us out of mental ruts, and providing healthy dose of distraction. I thank the universe for all of those things. More specifically, I could not have completed this work without the love and support of my family. They taught me that life is something that can be enjoyed. They also taught me that one can be both skeptical and optimistic, and that a good education is one of the most reliable means of realizing those ideas. Later, it was their encouragement that led me to pursue an advanced degree and helped me to work through times of doubt. I thank them for everything.

I would like to thank my advisor, Dr. Betty Cheng, for her thoughtful comments, constructive criticism, ninja-like editing skills, and patience.

I have a wonderful group of friends. Their indifference toward the day-to-day difficulties of my research served as a reminder that there is life outside of graduate school. I thank all of them for being supportive and fun to be with.

Two of the local public radio stations, WCMU and WKAR, kept the stream of jazz and classical music flowing when I needed it most, and they provided a valuable window into the goings on across the world. Their contribution to my cultural education and general well-being is deeply appreciated.

Finally, I want to thank my wife, Jenny. She has helped me every step of the way: applying to graduate school, finding a research area, choosing to write a thesis, and staying on task until the work was finished. However, these things just scratch the surface of her contribution. She brings untold warmth and joy to my life, and for this I am truly blessed.

TABLE OF CONTENTS

List of Tables . . . . . **vi**

List of Figures . . . . . **vii**

1 Introduction **1**
- 1.1 Thesis Statement 4
- 1.2 Thesis Contributions 4
- 1.3 Organization of Thesis 5

2 Background **6**
- 2.1 Unified Modeling Language (UML) 6
- 2.2 Refactoring 7
- 2.3 Design Patterns 8
  - 2.3.1 Definition and Motivation 8
  - 2.3.2 Terminology 9
  - 2.3.3 Example 10
  - 2.3.4 Pattern Descriptions 13
  - 2.3.5 Design Pattern Challenges 21
- 2.4 Evolutionary Computation 22
- 2.5 Metrics 24
- 2.6 Summary 28

3 Related Work **29**
- 3.1 Refactoring 29
- 3.2 Search-Based Refactoring 33
- 3.3 Design Pattern Detection 35
- 3.4 Automated Design Pattern Instantiation 37
- 3.5 Summary 40

4 Approach **42**
- 4.1 Overview 42
- 4.2 Individual Representation 43
  - 4.2.1 Design Graph 45
  - 4.2.2 Transformation Tree 45
- 4.3 Design Change Mechanisms 46
  - 4.3.1 Helper Functions 47
  - 4.3.2 Transformation Nodes 48
  - 4.3.3 Information Nodes 57
  - 4.3.4 Lazy Node Evaluation 58
- 4.4 Evaluating Refactored Designs 59
  - 4.4.1 Design Quality 59
  - 4.4.2 Design Pattern Detection 60

4.4.3 Fitness Function 61
4.5 Example 63
4.6 Summary 65

5 Validation 66
5.1 Prototype Implementation 66
5.1.1 Importing UML Class Diagrams 67
5.1.2 Representing the Target Software Design as a Graph 67
5.1.3 Mutation Operators for Manipulating the Graph 68
5.1.4 Implementing the QMOOD Metrics Suite 72
5.1.5 Detecting Design Patterns with PROLOG and SQL 72
5.1.6 Providing a Domain-Specific Vocabulary 73
5.2 Run-time Environment and Parameters 74
5.3 Hypotheses 75
5.4 Experiments 75
5.4.1 Experiment I: No Rewards or Penalties (Baseline) 76
5.4.2 Experiment II: Penalizing Large GP Trees 78
5.4.3 Experiment III: Rewarding Evolution of Design Pattern Instances 83
5.4.4 Experiment IV: Rewarding Specific Sequences of Minitransformations 89
5.4.5 Experiment V: REMODD Case Study 94
5.5 Discussion 96
5.6 Summary 97

6 Conclusions and Future Work 98
6.1 Summary 98
6.2 Impact 100
6.3 Future Work 100

A Queries for Design Pattern Detection 103
A.1 PROLOG Queries 103
A.2 SQL Queries 106

B Transcript of the ReMoDD GP Run 111

C XMI Representation of REMODD Design 119

REFERENCES 240

LIST OF TABLES

2.1 QMOOD Metric Descriptions 26

2.2 QMOOD Quality Characteristics and Formulas 27

5.1 Experiment I: Evolved Design Pattern Instances 77

5.2 Experiment II, First Treatment: Evolved Design Pattern Instances . . 79

5.3 Experiment II, Second Treatment: Evolved Design Pattern Instances 80

5.4 Experiment II, Third Treatment: Evolved Design Pattern Instances . 81

5.5 Experiment II: Number of Transformation Nodes 82

5.6 Experiment III, First Treatment: Evolved Design Pattern Instances . 84

5.7 Experiment III, Second Treatment: Evolved Design Pattern Instances 85

5.8 Experiment III, Third Treatment: Evolved Design Pattern Instances . 86

5.9 Experiment III, Fourth Treatment: Evolved Design Pattern Instances 87

5.10 Experiment III, Fifth Treatment: Evolved Design Pattern Instances . 87

5.11 Experiment III: Models with Significant Increase in DP Instances . . 88

5.12 Experiment IV, First Treatment: Evolved Design Pattern Instances . 90

5.13 Experiment IV, Second Treatment: Evolved Design Pattern Instances 91

5.14 Experiment IV, Third Treatment: Evolved Design Pattern Instances . 92

5.15 Experiment IV, Fourth Treatment: Evolved Design Pattern Instances 93

5.16 Experiment IV, Fifth Treatment: Evolved Design Pattern Instances . 93

5.17 Experiment IV: Models with Significant Increase in DP Instances . . 94

5.18 Experiment V: Evolved Design Pattern Instances 96

A.1 SQL Queries for Design Pattern Detection 110

B.1 Role-playing Classes in Sample `Abstract Factory` Instance 117

LIST OF FIGURES

2.1 Sample UML Class Diagram 7

2.2 Class Diagram for `AutoMaker` Example 11

2.3 `AutoMaker` Class Diagram with Abstract Factory 12

2.4 ABSTRACT FACTORY Class Diagram 14

2.5 ADAPTER Class Diagram 15

2.6 BRIDGE Class Diagram 17

2.7 DECORATOR Class Diagram 18

2.8 PROTOTYPE Class Diagram 19

2.9 PROXY Class Diagram 20

2.10 Process Diagram for Genetic Programming 23

2.11 Example of Subtree Crossover 24

4.1 Data Flow Diagram of the Proposed Approach 43

4.2 GP Components and Relationships 44

4.3 `Abstract Access` Minitransformation 49

4.4 `Abstraction` Minitransformation 50

4.5 `Delegation` Minitransformation 52

4.6 `EncapsulateConstruction` Minitransformation 53

4.7 `PartialAbstraction` Minitransformation 55

4.8 `Wrapper` Minitransformation 56

4.9 PROLOG code for ABSTRACT FACTORY design pattern 61

4.10 Example Transformation Tree Execution 64

B.1 Sample `Abstract Factory` Design Pattern Instance 117

B.2 Transformation Tree for Sample `Abstract Factory` Instance 118

Chapter 1

Introduction

Building a coherent object-oriented design for a piece of software is a difficult, time-consuming task. Designing that software to be easily maintained and extended in order to satisfy new requirements is even more difficult, as it forces software engineers to consider not only the details of the solution space (e.g., how to build on-board software for automobiles) but also details of the problem space (e.g., how to manage a large and ever-changing number of `AutoPart` subclasses and their object instances). These difficulties led to the creation of two important concepts in software engineering. First, *metrics* enable a software engineer to evaluate various characteristics of design quality, such as flexibility or readability, in a mathematically rigorous fashion. Second, *design patterns* provide a context-driven solution template for solving design problems that occur frequently in large software projects. However, metrics and design patterns often require significant intellectual investment to understand and implement, thus limiting their adoption. This thesis addresses these concerns by introducing an approach, based on evolutionary computation, that automates both the use of software engineering metrics and the introduction of design patterns in order to automatically generate sets of refactoring changes to improve an existing software design. The approach is modular, thus enabling the reconfiguration or substitution of metrics and design patterns as appropriate for different

domains, design constraints, and performance needs. In addition, the output from the approach includes the specific set of refactoring steps to apply in order to improve the quality of a design as measured by metrics. This information facilitates both manual and automatic application of the design changes, and our approach defers this choice to the software engineer.

Prior approaches to automated design improvement have focused primarily on automated *refactoring*, i.e. making small, behavior-preserving changes to software designs or source code, and have treated the problem as a pattern-matching activity. Ó Cinnéide and Nixon [28], as well as Sunye *et al.* [43], introduced sets of coarse-grained refactorings that, under "green-field conditions", can be applied automatically n order to improve design quality. However, the rule-based nature of these approaches limits their general applicability. In contrast, evolutionary computation (EC) harnesses the exploration and exploitation capabilities of Darwinian evolution to find the best solutions to an optimization problem [10], including many software engineering problems [8]. O'Keeffe and Ó Cinnéide [30,31] introduced the idea that an EC-based approach could recommend a set of fine-grained refactorings to apply in a given situation. They demonstrated that this approach is generally applicable and more flexible than prior refactoring work, suggesting that automated refactoring techniques are realizable. Seng *et al.* [40] proposed a similar approach that used a genetic algorithm to optimize the class hierarchy of an object-oriented program by moving methods from one class to another and evaluating the modified designs using metrics from Briand's catalogs [5]. While these approaches have demonstrated that EC-based techniques for refactoring are viable, they use simple refactoring steps as design mutation operators that result in small, incremental design changes. This limitation makes it difficult to compose multiple refactoring steps, a process that is necessary to facilitate complex design changes, such as the introduction of design patterns.

In contrast, this thesis introduces an approach to make the introduction of design patterns a first-class part of the refactoring process. This shift in focus carries three key benefits. First, we leverage the widely accepted notion that application of design patterns improves the reusability and maintainability of non-trivial software designs [38]. Second, evolutionary computation enables us to make use of transformations to generate more innovative refactoring solutions that incorporate design patterns. Third, the approach illustrates the use of a modular framework for search-based refactoring whose components (i.e., metrics, design patterns, and refactoring steps) can easily be substituted as appropriate for a given software project. By automatically constructing solutions that are highly rated by software engineering metrics and also contain design patterns, we reduce the manual, trial-and-error refactoring labor for software developers and also provide automatic design advice that is tailored to the software design in question. All things being equal, those refactoring solutions requiring the fewest transformations are favored. Furthermore, our approach generates the sequence of steps by which the input design can be transformed into an improved design, thus supporting the integration of this approach into automated design refactoring tools.

We use genetic programming (GP) [20], an evolutionary computing technique, to identify the most suitable set of refactorings to improve software design. As with any evolutionary approach, GP requires three components: a solution representation, a mechanism for making changes to that solution, and finally a means of measuring the solution's quality. Our solution representation takes the form of an abstract syntax tree whose nodes consist of so-called minitransformations introduced by Ó Cinnéide [27], as well as a graph structure that represents the software design that is to be improved. The minitransformations, when composed, can create instances of design patterns. The minitransformations make specific changes to the software design and are defined as functions that accept inputs and return outputs. Thus, they can be

expressed naturally as nodes in an abstract syntax tree. In order to evaluate the quality of solutions that are produced by the GP, we leverage the QMOOD metric suite that combines a rich set of metrics for analyzing the complexity of object-oriented hierarchies, cohesion of classes, and so on. QMOOD was introduced by Bansiya and Davis [2] and has been used in prior search-based refactoring work [31].

We illustrate the efficacy of this approach in a series of experiments by applying it to a large set of published models [14]. First, we demonstrate the approach without the use of specific rewards or penalties, thus allowing metrics alone to guide evolution. Second, we examine the benefits of applying a penalty in proportion to the number of refactoring steps suggested in order to limit the number of such steps that must be performed by hand. Third, we explore the benefits of explicitly rewarding individuals that evolve at least one design pattern instance. Fourth, we explore the benefits of rewarding the use of sequences of minitransformations that are known to construct design pattern instances. Finally, we apply the approach to a real-world case study [11].

1.1 Thesis Statement

It is possible to use techniques from evolutionary computation, with input and guidance from software engineering metrics, in order to improve the design of existing software through the introduction of design pattern instances.

1.2 Thesis Contributions

This thesis makes three contributions. First, we developed a novel genetic programming encoding that facilitates the automated refactoring of UML class diagrams. This provides a valuable proof of concept step that further supports the use of evolutionary computing to automate common software engineering tasks. Second, we

developed a process, guided by software engineering metrics, for introducing instantiated design patterns into existing software designs. Finally, the results from our approach provide traceability in the form of a step-by-step process for realizing the suggested refactoring sequence.

1.3 Organization of Thesis

The remainder of the thesis is organized as follows: Chapter 2 describes the background concepts of this work, including evolutionary computation, software metrics, and design patterns. Chapter 3 gives an overview of related work. Chapter 4 describes our approach for automated design improvement through the introduction of design patterns. Chapter 5 presents five experiments that validate our approach and document our exploration of the available parameters in our approach, thus providing a basis for choosing appropriate parameter values. The chapter also includes a case study that applies the proposed approach to a real-world web application. Finally, our conclusions and future work are presented in Chapter 6.

Chapter 2

Background

This chapter presents background concepts related to the approach that is proposed in this thesis. We begin with an overview of the Unified Modeling Language (UML), followed by a review of approaches to refactoring software designs. Next, we present the basic concepts of evolutionary computation. Finally, we discuss the measurement of quality in object-oriented software design using metrics.

2.1 Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a standardized, general-purpose modeling language that is designed to be used in the software engineering field. It defines a set of graphical notations that can be used to create visual models of software systems. In this thesis, we focus on UML *class diagrams*, which are box-and-line diagrams that depict the object-oriented structure of a software system, including the system's classes, their attributes and operations, and the associations between classes. A sample class diagram is depicted in Figure 2.1. In the sample diagram, there are five classes represented: `Drawing`, `Shape`, `Circle`, `Rectangle`, and `Triangle`. The latter three classes inherit the functionality (i.e., attributes and operations) of the `Shape` class. The inheritance relationship is indicated by the arrows with un-

shaded, triangular tips. The `Drawing` class *aggregates* many `Shape` objects, and this relationship is depicted by the arrow that has a diamond shape at its tail adjacent to the aggregating class. This aggregation relationship also specifies multiplicity constraints. The numeral 1 next to the `Drawing` class indicates that there is one `Drawing` instance in the relationship, and the asterisk (*) next to `Shape` indicates that there can be zero or more `Shape` instances in the aggregation.

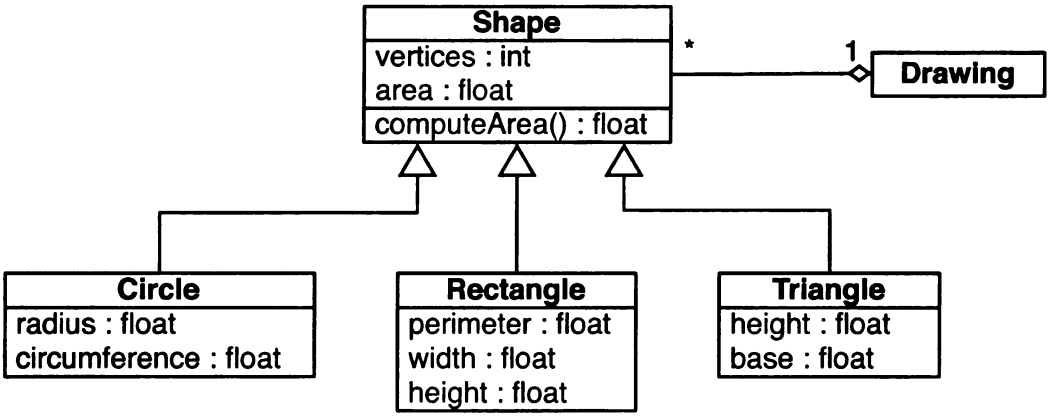


Figure 2.1: Sample UML Class Diagram

2.2 Refactoring

As software ages, its design will need to be refined as defects are discovered and requirements change. To address these changes, software developers make small, behavior-preserving changes called *refactorings*. These changes typically have no effect on a program's external behavior (i.e., its output) but instead focus on improving the program's design by making it more intuitive, more easily extended, or more maintainable [34].

Refactoring has been a target for automation, as evidenced by the large number of common refactorings that are included in popular software development tools such

[7]This page is dedicated to Jenny, for all of her love and support.

as Eclipse [9]. For example, Eclipse has a "rename class" refactoring that renames a class and updates all references to that class to be consistent with the new class name. This tool support reduces the number of tedious but important changes that software developers must perform manually. While industrial tools have focused on refactoring software at the source code level, techniques have been proposed that facilitate refactoring at the design, or model, level. Mens [23] proposed to model a software design as a graph and defines a set of refactorings in terms of graph transformations. Similarly, Marković [22] proposed a technique for refactoring Unified Modeling Language (UML) class diagrams annotated with constraints that specify pre- and post-conditions for each class. Focusing on software design in this way enables us to see patterns in its structure, and these patterns may reveal opportunities for refactoring steps that improve the software's modularity and long-term maintainability.

2.3 Design Patterns

In this section, we provide an overview of *design patterns*, beginning with a definition and motivating context. Next, we present an example illustrating the use of the `Abstract Factory` design pattern in a simple software system. We conclude with a textual description and graphical illustration of each design pattern that is supported in the proposed approach.

2.3.1 Definition and Motivation

In the field of software engineering, a *design pattern* is a general, reusable solution to a reoccuring problem in software design. It describes the core of a solution to the problem, deferring the specific implementation details to the developer [12]. This generic nature makes design patterns easier to understand and communicate than

implementation- or domain-specific solutions. Much like a document template, a design pattern gives developers a validated, robust starting point from which they can build a solution that suits their application and goals.

It is generally accepted by software engineers that software built using design patterns is more maintainable and extensible than software that is developed in an ad-hoc fashion without such established building blocks [38]. Since well-known design patterns are derived from developers' experiences over many years, it follows that design patterns are the manifestation of both long-term testing and collective experience. Thus, there is a clear advantage in using design patterns when they are relevant to a problem that a developer is facing.

More specifically, design patterns are intended to make existing software more flexible. In this thesis, we define *software flexibility* to be "the ease with which software can be modified or extended to include new functionality". Clearly, software flexibility directly affects the costs of maintenance and the feasibility of code reuse. Software that is not designed with flexibility and extension in mind will likely require greater effort and training to maintain during its lifetime than software that has been carefully designed to be amenable to future changes. As a result, much effort has been spent both in academia and in the software industry to develop tools and techniques that encourage flexible design concepts. Design patterns are one such tool, and their use has been widely cited as a good software engineering practice.

2.3.2 Terminology

In this thesis, we define the new term *design pattern template* to mean the generic description of a design pattern as given by Gamma *et al.* [12]. Each design pattern template contains a short name for the design pattern, a textual description of its purpose and the context in which it should be used, as well as an illustrative example. A design pattern template contains enough information that a software engineer will,

after reading it, understand the problem that the design pattern intends to solve and how to implement it.

In object-oriented software design, the process of creating an object instance from a given class is known as *instantiation.* Similarly, in this thesis we will refer to the process of creating a specific design pattern instance from a design pattern template as *design pattern instantiation.* Just as an object instance is a more concrete version of a class, a design pattern instance is a more concrete, application-specific version of a design pattern template.

2.3.3 Example

Design patterns can be difficult to present or understand from a simple text description. In this section, we present an example that illustrates the application of the `Abstract Factory` design pattern to an existing design.

First, we introduce a software design called `AutoMaker` that can benefit from the introduction of a design pattern. `AutoMaker` represents a software system for an automobile manufacturer. A partial UML class diagram for `AutoMaker` is shown in Figure 2.2. In `AutoMaker`, there are three classes that represent the style of automobiles to be manufactured. These include `Convertible`, `Coupe`, and `Sedan`. Each of these classes has a corresponding factory class, namely, `ConvertibleFactory`, `CoupeFactory`, `SedanFactory`. These factories construct instances of the automobiles, and the instances are then shipped by a freight company (represented by the `FreightCompany` class) to the customer.

While this seems to be a reasonable software design, it has a flaw that becomes apparent when the automobile manufacturer adds another vehicle to its product line. In order to introduce a compact vehicle style, we must introduce a `Compact` class. Next, we must introduce a `CompactFactory` class in order to make instances of `Compact`. Finally, the `FreightCompany` class must be modified in order to ship

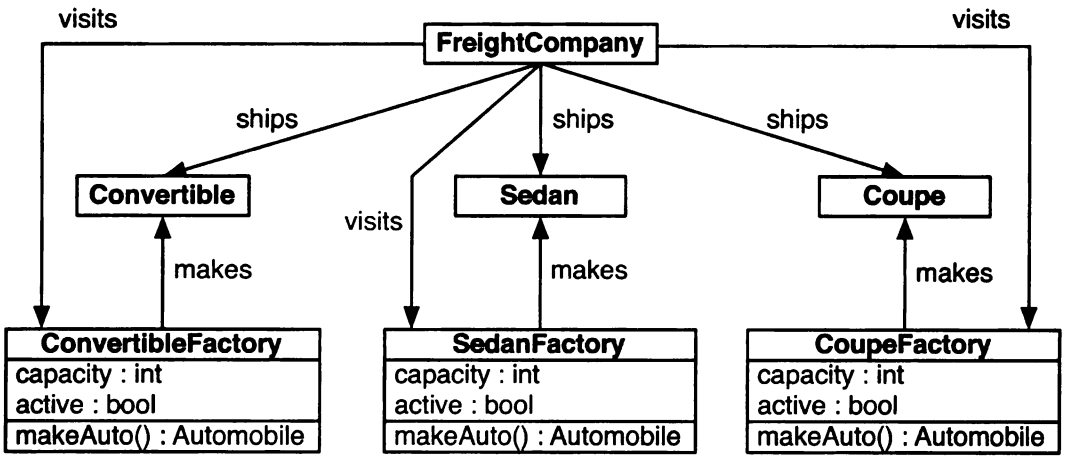


Figure 2.2: Class Diagram for `AutoMaker` Example

instances of `Compact`. In an ideal design, the `FreightCompany` does not need to know when the automobile manufacturer has introduced a new product. That is, it is designed according to a common interface that is shared by all automobiles. Similarly, we observe that the factory classes duplicate much of their functionality when new automobile styles are introduced. This duplication is another scalability problem that can be resolved using object-oriented techniques.

We now consider how to modify the software design to address the two flaws mentioned above. First, to enable the freight company to take a more abstract view of the automobiles that they are shipping, we introduce an abstract class named `Automobile`. We then modify the `Convertible`, `Coupe`, and `Sedan` classes so that they inherit `Automobile`. Now that we have a common interface for all automobiles, we modify the `FreightCompany` class so that it ships instances of the abstract class `Automobile`. These changes are illustrated in the top half of Figure 2.3.

Next, we consider the duplication of functionality among the factory classes. Since the three factory classes in the original class diagram have common attributes (i.e., `capacity` and `active`) and a common operation `makeAuto`, it is clear that a second abstract class is needed. Therefore, we introduce an abstract `Factory` class that

comprises the same attributes and operation as the original factory classes. Next, we modify the factory classes to inherit `Factory`, thus eliminating the duplication. The resulting class diagram is given in Figure 2.3.

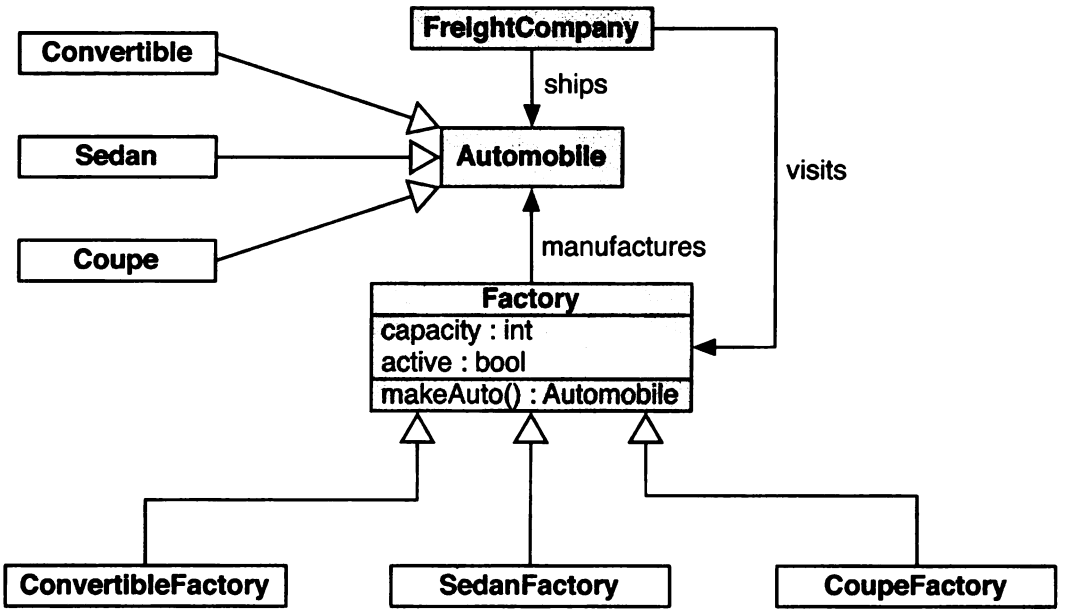


Figure 2.3: `AutoMaker` Class Diagram with Abstract Factory

In the course of this example, we have constructed an instance of the **`Abstract Factory`** design pattern. The purpose of **`Abstract Factory`** is to create a layer of abstraction between the *product* classes and the *factory* classes that create instances of those products. This layer ensures that the product classes, and especially the client classes, can vary independently of the factory classes, thus reducing the accidental complexity [6] of the software design as it changes over time in response to new requirements. The pattern has three roles: *Abstract Factory*, *Abstract Product*, and *Client*. In this example, those roles are fulfilled by **`Factory`**, **`Automobile`**, and **`FreightCompany`**, respectively. The classes filling these roles are shaded in Figure 2.3.

2.3.4 Pattern Descriptions

We now present detailed descriptions for each of the design patterns that are used in this thesis. These descriptions are derived from *Design Patterns: Elements of Reusable Object-Oriented Software* [12], in which Gamma *et al.* define three categories for their design patterns: *behavioral* patterns that focus on common communication patterns between objects and class, *creational* patterns that solve common problems related to creating objects, and *structural* patterns that identify reusable ways to realize relationships between entities (e.g., objects and classes). There is a great deal of diversity among the Gamma design patterns in terms of their level of detail. Due to the nature of our approach, we restrict the usable set of patterns to a subset of the *creational* and *structural* pattern categories. These patterns are identifiable based on their footprint in UML class diagrams, thus enabling us to detect their instances when they happen to evolve in a GP.

Abstract Factory

A *factory* is a class that is responsible for creating instances of other classes. In a drawing program, for example, the `PixelFactory` class could create instances of the `Pixel` class that are ultimately rendered to the screen. The `Abstract Factory` pattern provides a way to encapsulate a group of individual factories that have common functionality. A graphical depiction of `Abstract Factory` is shown in Figure 2.4. The pattern calls for the creation of an abstract `Factory` class that defines the interface for all other factories. Classes that need to use a factory, then, can be written to the interface. At runtime, these classes can utilize concrete factory classes without any knowledge of their specific implementation.

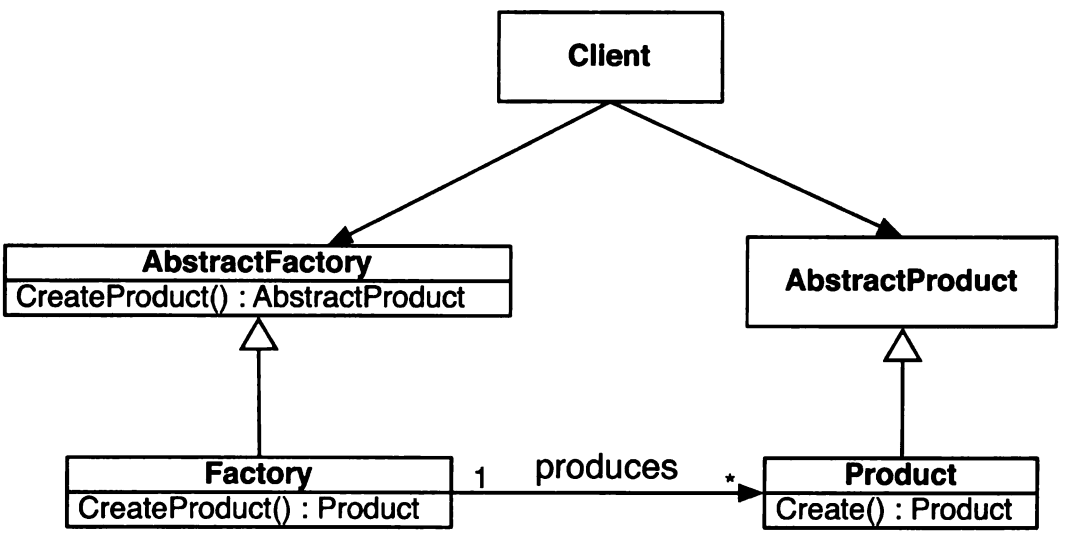


Figure 2.4: ABSTRACT FACTORY Class Diagram

Adapter

Two or more classes that were designed independently can be difficult to combine. The classes might make incompatible assumptions about their inputs or outputs, and modifying the classes to be compatible with one another might be infeasible. The `Adapter` design pattern translates, or adapts, the interface for a class into an interface that is compatible with another class. A graphical depiction is shown in Figure 2.5.

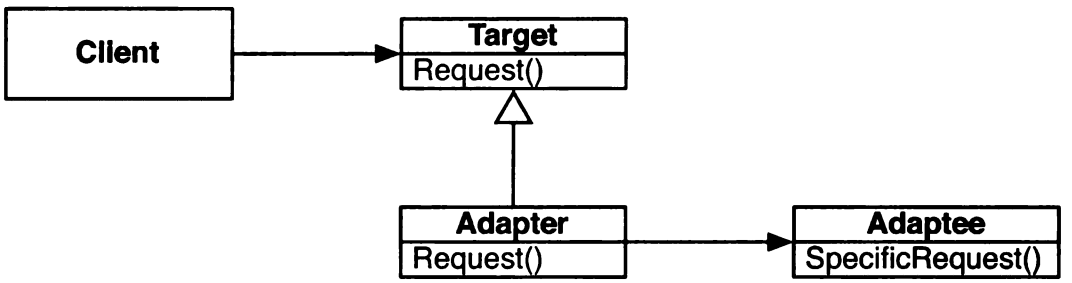


Figure 2.5: ADAPTER Class Diagram

Bridge

In many software systems, multiple implementations exist for a given abstraction. Object-oriented design supports this scenario with *inheritance*, which enables a class to take on the attributes and operations of a parent (inherited) class. However, situations arise where this tight binding between a concrete implementation and its abstract inherited class is too restrictive. One solution to this problem is the `Bridge` design pattern. This pattern is meant to decouple an abstraction from its implementation so that the two can vary independently [12]. `Bridge` is depicted graphically in Figure 2.6.

To motivate the use of the `Bridge` design pattern, we consider the Windowing toolkit example presented by Gamma *et al.* [12]. A `Window` abstraction in a user interface toolkit may need to be deployed on multiple platforms. Specifically, it needs to be able to display `Window` objects on the X Window System on Linux [47] as well as the Aqua windowing system on Apple Mac OS X (TM). A first approximation of a solution to this problem is to use inheritance; that is, we should create a concrete subclass of `Window` for each of the windowing systems. However, this solution has two major drawbacks. First, if we wish to extend the functionality of `Window` to include support for other types of windows, then we must re-implement each of the concrete subclasses. Second, the solution requires that a client class (i.e., a class that creates a `Window` instance) commit itself to a specific `Window` implementation. To address these problems, the `Bridge` design pattern separates the `Window` abstraction and classes that implement it into separate class hierarchies. This solution enables us to extend the `Window` abstraction to include new window types while reusing the existing concrete implementation classes. In addition, the clean separation of the class hierarchy enables client classes to view the hierarchy more abstractly, and therefore resolves the drawback of clients committing to a specific concrete implementation.

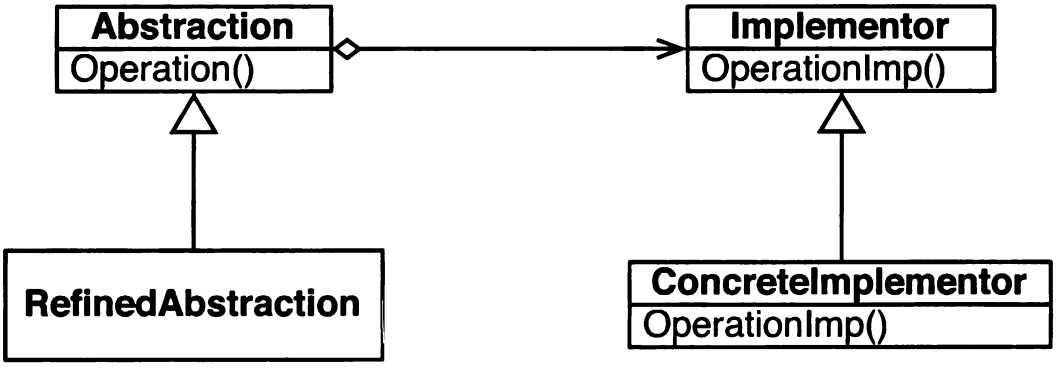


Figure 2.6: BRIDGE Class Diagram

Decorator

The `Decorator` pattern is a design pattern that allows new or additional behavior to be added to an existing object dynamically. `Decorator` is depicted graphically in Figure 2.7. For example, a Windowing toolkit might wish to add borders to a specific object instance of the `Window` class. `Decorator` accomplishes this change in functionality by wrapping, or enclosing, an object with another (decorator) object. The enclosing object conforms to the interface of the enclosed object, thus permitting it to interact transparently with clients of the enclosed object.

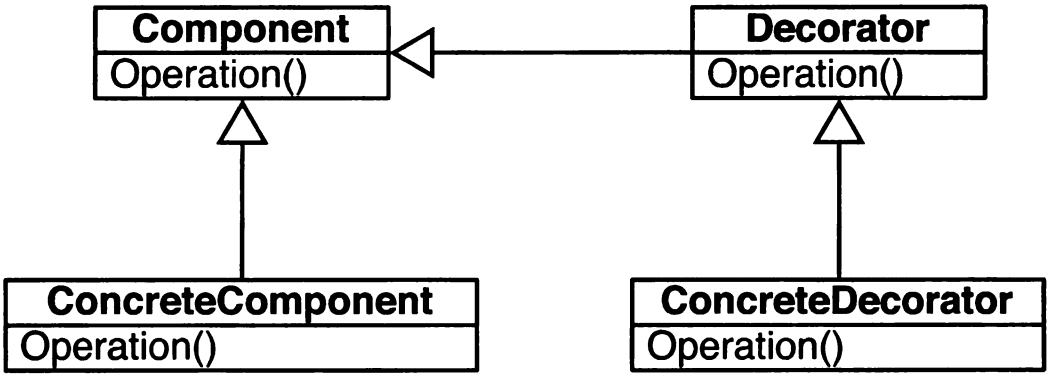


Figure 2.7: DECORATOR Class Diagram

Prototype

The `Prototype` design pattern is used when many objects need to be created by cloning a prototypical object instance. `Prototype` is depicted graphically in Figure 2.8. This design pattern avoids the need for subclasses, thus making it different from the `Abstract Factory` pattern. It also offers a solution for scenarios in which it is prohibitively expensive to create new object instances using, for example, the `new` keyword.

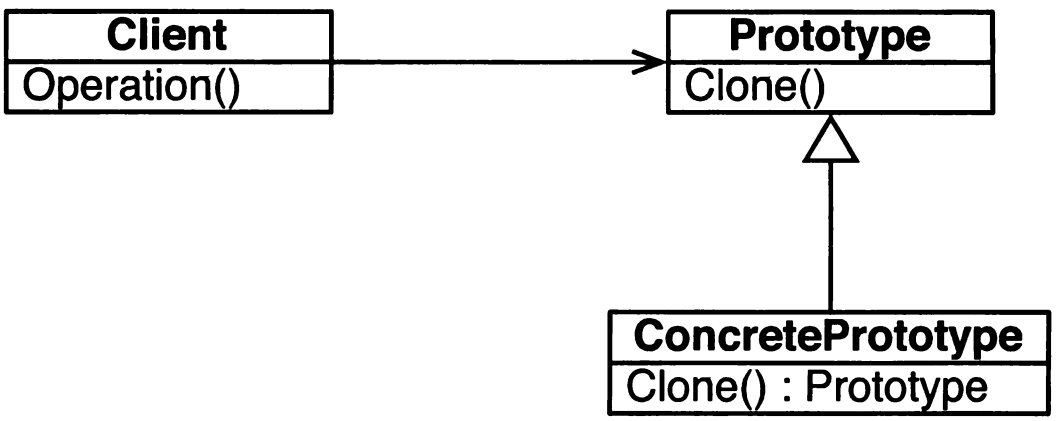


Figure 2.8: PROTOTYPE Class Diagram

Proxy

The **Proxy** design pattern comprises a class functioning as an interface to something else. **Proxy** is depicted graphically in Figure 2.9. This proxy class can be an interface to any system element: a network connection, a large object in memory, a file, or some other resource that is expensive or impossible to duplicate. For example, loading a large file into memory is an expensive operation. Using the **Proxy** design pattern, one can load the file into memory once and then provide a light-weight proxy object that has the same interface as the file to each class that wishes to use it. Once all of the proxy objects are destroyed (e.g., from passing out of scope), the large file can be closed and purged from memory.

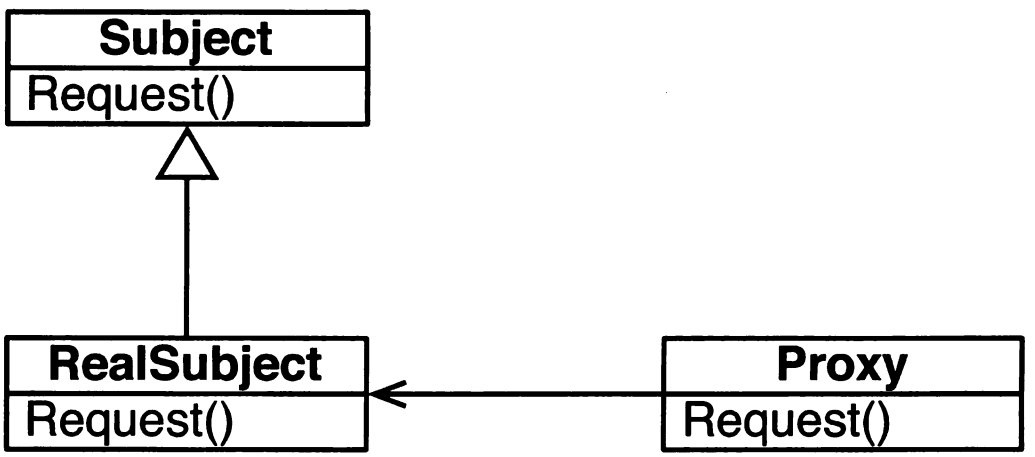


Figure 2.9: PROXY Class Diagram

2.3.5 Design Pattern Challenges

In this thesis, we leverage the widely accepted notion that application of design patterns improves the reusability and maintainability of non-trivial software designs [38]. However, despite widespread adoption and general recognition that their use encourages more reusable and maintainable software design, design patterns have received criticism. We discuss several such criticisms in this section.

Documenting design pattern use is recognized as a difficult problem [46]. Applying a design pattern that involves multiple classes causes changes to each of those classes, which makes their documentation difficult to maintain in a single place. A developer can place a note in the software's high-level design comments, but including design pattern-related comments in code is also important for code readability. When multiple design patterns are applied to a single software system, the documentation problem is sometimes called the design pattern *composition problem* or *overlapping problem* [17].

In response to the design pattern documentation problem, He *et al.* [17] developed a UML-based method of analyzing design pattern composition and tracing the effects of that composition. Their work enables developers to view a software model as a composition of design patterns and enable that model to be viewed at different levels of abstraction.

Schulz *et al.* [39] identifies two major problems with design patterns: first, there is no formal classification for design patterns and their mutual relationships. Second, there has been limited effort toward a systematic way to integrate design patterns into existing systems [51]. We address the second problem in this thesis.

2.4 Evolutionary Computation

Evolutionary computation (EC) is a branch of machine learning that harnesses the concepts of natural (Darwinian) evolution in order to evolve computer programs that solve optimization problems [10]. Figure 2.10 illustrates the problem-solving flow of an EC technique. EC techniques comprise a population of initially-random *individuals.* Each individual encodes a solution representation, usually in the form of a bit-string or an abstract syntax tree. These solutions are evaluated according to a *fitness function*, a mathematical formula that measures an individual's ability to solve a given problem. The individuals are then selected for placement into a subsequent population (the next generation) based on their fitness. Selection is typically performed using an algorithm known as *tournament selection.* This algorithm draws a sample of n individuals from the population, compares their fitnesses, and selects the individual with the highest fitness. The value n is called the *tournament size* and is typically less than 10. After being selected, these individuals undergo a random mutation and, through a process known as *crossover*, portions of their solution representations are recombined with solutions from other individuals in order to explore a greater portion of the solution space. This process of fitness measurement, selection, and mutation is continued for a fixed number of generations or until the optimal solution to the problem being solved is discovered.

One EC technique, known as genetic programming (GP), uses an abstract syntax tree representation for the individuals in its population [20]. An abstract syntax tree (AST) can represent, for example, an algebraic equation. In this situation, the nodes of the AST represent arithmetic operations, at least one algebraic variable, and numerical values. A population of these individuals can be given the task of finding an algebraic function that most closely fits a finite set of two-dimensional data points. This is known as the *Regression Problem.* In the context of this problem, the fitness of an individual can be defined as the sum of the function's error (i.e.,

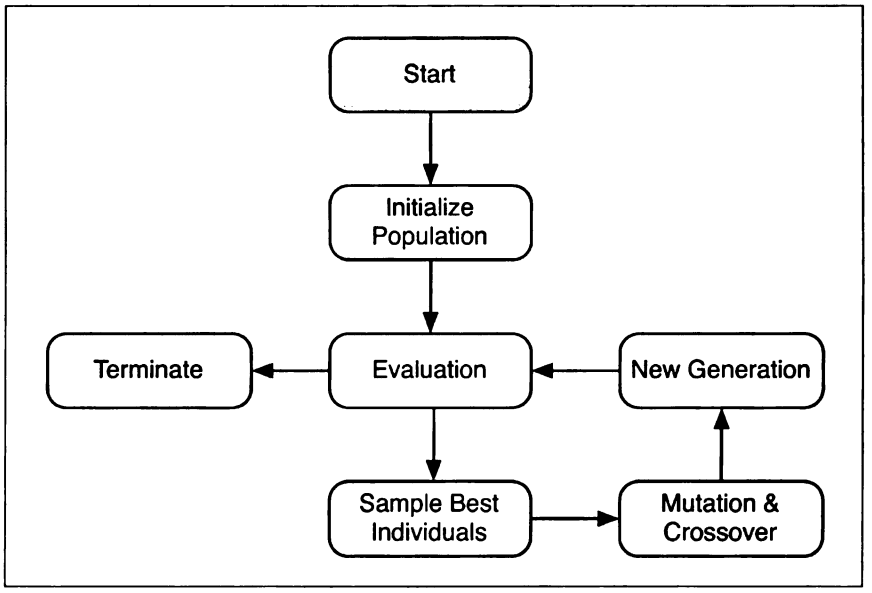


Figure 2.10: Process Diagram for Genetic Programming

the difference between the function value and the value of the corresponding data point) over the set of points. Intuitively, an individual is highly fit if its algebraic function very closely matches the set of points. In this section, we assume that the function $f(x) = x^2 + x$ is the optimal fit for the data points that have been given to the GP as input.

Beginning with a population of random individuals, each representing an algebraic function, the GP uses tournament selection to select a sample of individuals that are highly fit relative to others in the population. After being selected, these individuals will be randomly mutated. This mutation could involve, for example, changing a node that represents a numerical (e.g., 7) into a node that represents an arithmetic operator or an algebraic variable. If the new node is an arithmetic operator that requires child nodes in order to be meaningful, then these nodes are randomly grown.

After being mutated, individuals also undergo *subtree crossover*. This process is illustrated in Figure 2.11. Subtree crossover swaps subtrees between the ASTs of two parent individuals, thus creating two child individuals that reuse portions of

their parents' ASTs but are also structurally different from them. In Figure 2.11, the subtrees to be swapped are shaded in the AST of parent nodes, and the arrows identify the new location of those subtrees in the child nodes. At this point, both child individuals are inserted into a new population that will become the next generation. We note that `Child 1` in Figure 2.11 represents the function $f(x) = x^2 + x$, and thus it is the optimal solution. When the GP evaluates the next generation, this individual will be marked as *optimal*, and the GP will terminate.

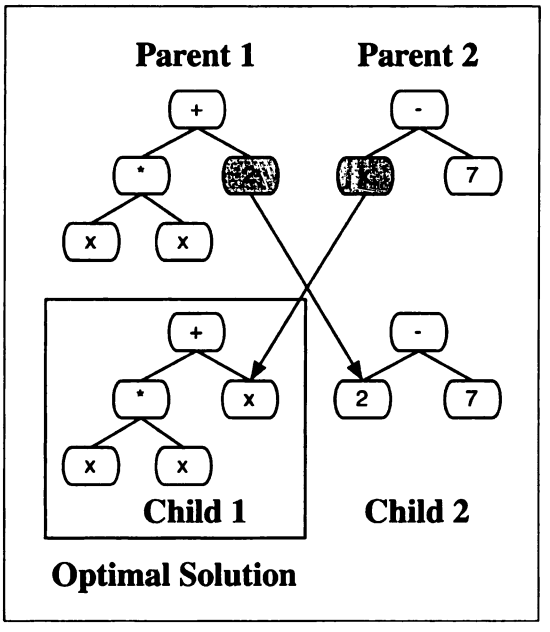


Figure 2.11: Example of Subtree Crossover

The combination of mutation and subtree crossover that we described in this section has been shown to explore efficiently, over the course of several generations, a large space of candidate solutions. In this paper, we use genetic programming as the basis for our approach to automated improvement of software design.

2.5 Metrics

As the word suggests, a *metric* is a mechanism for measuring a specific attribute of an element, with a specific focus on how that attribute changes over time. Many

software engineering metrics have been proposed in the literature [2, 5, 7, 18, 36]. These metrics measure aspects of software design from the number of classes, the average size of an interface, and so on. They provide both an instantaneous snapshot of these characteristics and, when applied over time, a profile of how a software design has changed through its lifetime.

The Hierarchical Metrics for Object-Oriented Design Quality (QMOOD) suite introduced by Bansiya and Davis [2] comprises 11 individual metrics, each of which evaluates a distinct aspect of object-oriented design quality. Notably, these metrics are designed to be amenable to automated evaluation, thus making them ideal for use in software engineering tools. From these metrics, Bansiya and Davis derived a set of formulas that measure abstract quality characteristics such as extensibility, readability, and maintainability. These high-level quality characteristics can inform software developers of the ways in which their software is well-designed and, conversely, where it can be improved. Additionally, the QMOOD authors recommend that a real-valued weight should be assigned to each characteristic in order to specify its relative importance, thus providing automated feedback regarding how well a software design meets that goal.

A complete listing of the metrics in our QMOOD implementation is given in Table 2.1, adapted from O'Keeffe and Ó Cinnéide [32]. We implemented each of these metrics as a graph-based algorithm that analyzes specific characteristics of a target software design, such as the number of disjoint inheritance hierarchies. Each metric returns a floating-point number, and those numbers are used as input for formulas that evaluate complex quality characteristics. These characteristics, and the formulas that are used to compute them, are shown in Table 2.2.

Metric Name	Description
Design size in classes (DSIC)	The number of classes in the software design.
Number of hierarchies (NOH)	The number of class hierarchies in the software design.
Average number of ancestors (NOA)	The average number of other classes that a class inherits.
Number of polymorphic methods (NOPM)	The number of methods in the software design that exhibit polymorphic behavior.
Class interface size (CIS)	The average number of public methods in a class.
Direct class coupling (DCC)	A count of the number of classes that a given class is directly related to by attribute declaration or method return type.
Cohesion among methods of class (CAM)	The relatedness among methods of a class, computed using the summation of the intersection of parameters of a method with the maximum independent set of all parameter types in the class.
Number of methods (NOM)	The average number of methods in a class.
Data access metric (DAM)	The ratio of non-public (i.e., private or protected) attributes to the total number of attributes declared in a class. This is interpreted as the average of the ratios for all classes in the software design.
Measure of aggregation (MOA)	The average number of data declarations (e.g., fields) in a class whose data types are user-defined classes. We exclude classes that are part of the Java standard library and language.
Measure of functional abstraction (MFA)	The ratio of the total number of methods inherited by a class to the number of methods that are accessible by member methods of that class.

Table 2.1: QMOOD Metric Descriptions

Characteristic	**Formula**
Reusability	$0.5 * DSIC - 0.25 * DCC + 0.5 * CIS$
Flexibility	$0.25 * DAM - 0.25 * DCC + 0.5 * MOA + 0.5 * NOPM$
Understandability	$-0.33 * DSIC - 0.33 * NOA + 0.33 * DAM - 0.33 * DCC - 0.33 * NOPM - 0.33 * NOM$
Functionality	$0.22 * NOPM + 0.22 * CIS + 0.22 * DSIC + 0.22 * NOH$
Extendibility	$0.5 * NOA - 0.5 * DCC + 0.5 * MOFA + 0.5 * NOPM$
Effectiveness	$0.2 * NOA + 0.2 * DAM + 0.2 * MOA + 0.2 * MOFA + 0.2 * NOPM$

Table 2.2: QMOOD Quality Characteristics and Formulas

2.6 Summary

In this chapter we presented background concepts related to the approach that is proposed in this thesis. First, we gave an overview of the Unified Modeling Language (UML), followed by a discussion of approaches for refactoring software designs. Next, we presented the basic concepts of evolutionary computation. Finally, we discussed the measurement of quality in object-oriented software design using metrics.

Chapter 3

Related Work

In this chapter, we present a review of the literature related to the approach that is proposed in this thesis. We begin with a discussion of prior work related to refactoring. Next, we discuss work that has used evolutionary techniques to perform refactoring tasks. Third, we review techniques for detecting design pattern instances in software designs. Finally, we discuss methods for automatically instantiating design patterns in existing software designs.

3.1 Refactoring

Since the requirements of existing software systems tend to change over time, there is a strong need for tools and techniques that make software maintenance more efficient and, when possible, more automated. One promising concept is that of *refactorings*, which are *organization plans that support change at an intermediate level and do not change the behavior of a program* [34]. Change at an "intermediate level" means that refactorings focus on changes that are more abstract than rewriting lines of code but more concrete than adding features to a system. For example, a refactoring might involve moving an operation from one class to another. This change might make the program more streamlined or easier to understand, but it will not affect the

program's behavior. This property is known as *behavior preservation* (also known as *semantic equivalence*), and informally means that one could pass the same input to a program before and after the transformation is performed and receive precisely the same output from each. Studying the theoretical foundations of refactoring and establishing a formal basis for it has been a common subject of research in the software engineering field. We present a summary of such research in this section.

Behavior-Preserving Refactoring of Object-Oriented Programs. Opdyke developed a set of refactorings [34] that are intended to facilitate behavior-preserving modification of object-oriented code, with a specific focus on encouraging the development of automated refactoring tools. These transformations are specifically designed to make the program more maintainable, flexible, or extensible, while respecting behavior preservation. Behavior preservation for each refactoring is enforced by pre-conditions and post-conditions that must be met before and after the refactoring is performed, respectively. As a simple example, a post-condition might be the following: "After the refactoring is applied, each class must have a unique name".

OCL-Based Specifications. A similar set of refactorings were proposed in 2001 by Sunye *et al.* [43] that make use of the Object Constraint Language (OCL) [50] as a formal foundation. OCL is a declarative language that supports the specification of pre- and post-conditions on elements in the UML meta-model. For example, one could use OCL to specify that a method *GetAutoID* must exist in a class *Automobile* before a refactoring is performed (a pre-condition) and afterward (a post-condition). The authors provide sample OCL code for refactorings that deal with class diagrams and statecharts. More specifically, these OCL constraints are designed to supervise the addition, removal, relocation, and promotion (in the sense of generalization and specialization) of attributes and methods in class diagrams to ensure that model

consistency is preserved during the refactoring. For refactoring statecharts, the constraints support replacing a set of equivalent actions attached to transitions with an entry or exit action attached to the state, or, symmetrically, unfolding an entry or exit action into a set of actions attached to the transitions to and from a state. Furthermore, a state can be moved into a composite state in order to simplify an existing statechart.

Using Object-Oriented Metrics. Tahvildari and Kontogiannis [44] propose a methodology in which a catalog of object-oriented metrics is used to decide when a particular transformation should be applied in order to improve the quality of a software design. The work analyzes the interaction between specific transformations and object-oriented metrics. Metrics are also used to detect common design flaws and to decide when a corrective transformation should be performed. The paper classifies design flaws into three groups: structural flaws, behavioral flaws, and architectural flaws. These categories enable detection and correction of a given flaw based on its level of abstraction. From these three categories, a finer-grained classification system is also presented. Additionally, the authors define three categories of metrics: complexity, coupling, and cohesion. These metrics are derived from the literature on measurement of software quality. The proposed methodology was implemented as a Java tool and validated on Java Expert System Shell, a rule engine and scripting environment.

Refactorings as Graph Manipulations. Mens *et al.* proposed a formal model for refactoring based on graph transformations [24]. By representing the object-oriented class hierarchy of a program as a graph whose vertices are classes (or operations) and whose edges are relationships among those elements, the authors were able to utilize available research and tools for analyzing and manipulating graphs. As with the approaches already mentioned, this graph-based technique used pre- and

post-conditions in order to ensure preservation of the intent and, ideally, the behavior of a given program after it is refactored. The authors developed a notation similar to regular expressions that facilitates searching the graphs for patterns (e.g., to determine whether the graph contains a method invocation from method X to method Y). Using this notation, a software designer can specify rules (pre-conditions) that must be satisfied before a given refactoring can be performed. Likewise, the rules can verify that aspects of a program remain valid (post-conditions) after the refactoring is performed. Within this framework, the authors reduced the problem of refactoring to one of transforming an input graph into an output graph by way of a series of simple graph modifications whose validity can be verified.

Fixing Design Flaws. Trifu *et al.* developed a method for identifying and fixing design flaws using human-assisted tools based on mappings between common design flaws and solutions [48]. The authors focused on design flaws that affect the non-functional goals of the program (e.g., extensibility or flexibility). They define a *correction strategy* to be a mapping between a design flaw and the set of possible solutions (refactorings) that, if applied, would correct that flaw. Detection of design flaws is performed using graph-based pattern matching in a manner similar to that of Mens *et al.* discussed previously [24]. In order to determine the order of preference among the set of possible solutions, a formal grammar is defined that can be used to specify atomic code transformations along with their respective difficulty. The influence of each design flaw on non-functional goals (e.g., extensibility) is recorded as well. With this information in hand, the authors' prototype tool (known as the Advanced Refactoring Wizard) is able to make refactoring decisions automatically and will prompt the user only when the tool cannot reach a conclusion.

3.2

3.2 Search-Based Refactoring

Harman and Jones introduced the term *search-based software engineering* (SBSE) [15], arguing that the benefits of evolutionary computation could be brought to bear on software engineering problems that can be reformulated as optimization problems. Harman and Clark [16] presented evidence that existing object-oriented metrics could be used to construct effective fitness functions for evolutionary approaches to automated software maintenance and refactoring. Possible improvement mechanisms include popular evolutionary computing techniques, such as genetic algorithms, genetic programming, simulated annealing, and hill climbing.

O'Keeffe and Ó Cinnéide [30] considered the problem of automatically improving the design of an object-oriented program using a search-based approach. Calling the technique *search-based refactoring*, they selected simulated annealing (SA) as the improvement mechanism and implemented two complementary mutation operators that SA could use to modify a program. These operators were **`pullUpMethod`**, which moves a method to the superclass of the class in which it was defined; and **`pushDownMethod`**, which moves a method to the subclasses of the class in which it was defined. Next, the authors constructed a fitness function using two simple metrics: **`availableMethods`**, computed as the sum of the methods declared within a given class and the methods inherited by that class; and **`methodsInherited`**, computed as the number of methods that a given class inherits. By minimizing **`availableMethods`** and maximizing **`methodsInherited`**, a crude measure of quality could be determined for a given program. This quality was used as part of the simulated annealing process in order to determine how the program should be modified. The approach assumes that the source code of the program is available, and this code is parsed into an abstract syntax tree (AST) by a prototype tool. Refactoring steps are then performed on this syntax tree, and the resulting structure is evaluated according to the metrics described above. Using a simple case study that initially

contains code duplication and unnecessary methods, the authors showed that their automated approach could refactor a program so that its methods were positioned optimally in the hierarchy with respect to the metrics that were used to evaluate the quality of the program.

In subsequent studies, O'Keeffe and Ó Cinnéide considered a greater variety of evolutionary algorithm, including single- and multiple-ascent hill climbers and a genetic algorithm [33, 35]. Furthermore, the authors evaluated the evolving programs using a more sophisticated quality model known as Quality Model for Object-Oriented Development (QMOOD) [2]. QMOOD assesses the quality of a program's object-oriented design by examining interface size, inter-class coupling, and use of inheritance, among others. After examining each evolutionary algorithm on four case studies, the authors show that multiple-ascent hill climbers are more effective than other mechanisms in terms of speed and consistency of quality gain, but the advantage is not statistically significant. However, the authors do not consider the use of genetic programming, which is a cornerstone concept in our proposed approach and has the potential to generate richer, more complex refactoring strategies through composition of multiple design changes.

Seng *et al.* [40] propose a search-based refactoring approach that uses a genetic algorithm and a set of metrics from Briand's catalog [5] in order to optimize the class structure of an object-oriented program. The selected metrics evaluate coupling, cohesion, complexity, and stability. The approach considered only the movement of methods from one class to another, and it is therefore less general than the approaches of O'Keeffe and Ó Cinnéide. However, the authors pointed out the need to consider the *role* that a given class plays in a design. For example, design patterns will often violate design guidelines (e.g. cohesion) in order to fulfill their responsibilities, and `get/set` methods should typically remain in the classes in which they are defined. Using this information in a search-based technique would reduce the size

of the search space by removing uninteresting solutions. However, the authors defer the use of such heuristics to future work. To evaluate their approach, the authors selected the JHotDraw open source drawing framework, which is known as a good example for the use of design patterns. The authors manually displaced several methods from their original class in order to provide additional refactoring tasks for the approach to perform. The prototype tool that implements this approach moved each displaced method back to its original class at least once. Furthermore, the tool suggested several refactoring sequences that the authors deemed helpful. Compared to our approach, the proposed technique focuses exclusively on the relocation of classes and does not consider composition of refactoring steps, thus limiting the expressiveness of its solutions. However, the relocation of methods to optimize the class structure of a software design is a useful refactoring strategy.

3.3 Design Pattern Detection

In this section, we review approaches for automatically detecting instances of design patterns in source code and UML models. Unless otherwise specified, these approaches only consider the detection of Gamma design patterns [12].

Krämer and Prechelt [21] propose a PROLOG-based method for detecting instances of design patterns in object-oriented source code. The approach extracts design information from C++ header files and translates them into PROLOG facts (e.g., inheritance between class `A` and class `B` might be represented as "inherits A B"). Each design pattern is represented as a PROLOG *rule* or *query* that identifies every subset of facts that satisfies the terms of the query. In the context of design pattern detection, each PROLOG query searches for all combinations of classes, interfaces, or operations that can properly fill the roles of a particular design pattern. The result of each query, then, is the set of *candidate design pattern instances* in the software design. The authors implemented their approach in a tool called `Pat` and

demonstrated through an empirical study that while its recall (i.e., the tool's ability to detect known design pattern instances) was 100%, its precision (the number of correctly-detected design pattern instances divided by the total number of instances detected) ranged from 14% to 50% depending on the program being analyzed. The authors note that while this precision is "acceptable", it would be much higher (i.e., 53-100%) if the structural analysis phase were capable of detecting method call delegations. For each questionable design pattern instance, the authors consulted the program source code and its documentation in order to decide whether it is valid.

Birkner [4] proposes a similar method to that of Krämer and Prechelt that extracts design information from Java bytecode. Using a toolchain derived from the SWAG Kit [42], his approach uses a set of Query Language (QL) queries that analyze bytecode facts to identify candidate instances of *structural* design patterns. The structural queries presented are similar, though more coarse-grained, than the PROLOG queries of Krämer and Prechelt. Birkner's approach also supports the detection of *behavioral* design patterns. The approach instruments the Java bytecode and extracts a trace of the program's execution, thus enabling the analysis of runtime information that is not available from static, structural artifacts such as source code or UML class diagrams. The author experimentally demonstrates that his approach is able to detect the Gamma design patterns with more precision than PINOT [41] or FUJABA [25], which are design pattern detection tools.

Tsantalis *et al.* [49] propose the use of graph similarity scoring for detecting design pattern instances. The approach represents a software design as a graph (i.e., a set of vertices and a set of edges between those vertices), thus enabling the use of established graph (and matrix) analysis algorithms. Specifically, each type of association between classes is modeled as a distinct matrix. One matrix depicts inheritance relationships, another depicts method invocations, and so on. A set of these matrices is defined for each design pattern, as well as the target software design. From this

foundation, the authors define a *similarity matrix* that results from comparing the matrices for the target software design against the matrices for each design pattern. This process returns a *similarity score* within the range $[0, 1]$. A high similarity score (i.e., a value near 1) for a given design pattern indicates that an instance of that pattern probably exists in the target software design. Conversely, a low similarity score indicates a low probability that such a design pattern instance exists. This approach, in contrast to PROLOG based techniques, enables the detection of design pattern instances that differ from their standard representation. Given the nuanced nature of software design, this flexibility is a powerful asset. However, due to the computational complexity of large matrix operations, the approach requires that the target design be partitioned into *subsystems*, thus limiting its general applicability.

Our approach leverages the work of Krämer and Prechelt [21] to detect design pattern instances using PROLOG. We extend their work, as well as the work of Birkner [4], to detect design pattern instances more efficiently using the Structured Query Language (SQL).

3.4 Automated Design Pattern Instantiation

In this section, we review work that is most similar to our proposed approach. These techniques focus on the automatic instantiation of design patterns, either in program source code or in UML models.

Design patterns have historically been treated as building blocks that are most effective when a new software system is being constructed. However, Schulz *et al.* [39] recast design patterns as *operators* that transform an existing (input) software design into a target (output) design. The approach uses Opdyke's refactorings [34] as a basis for the transformation. After a successful application of a given operator, the target design contains an instance of the design pattern that corresponds to the operator. The authors proposes a five-step process for using the operators. It is

assumed that a software design exists and that a software engineer wishes to apply a design pattern to that design. First, a software engineer identifies the portion of the software design that should be reorganized and selects the design pattern operator that will perform the reorganization. Second, the preconditions associated with the design pattern operator are checked to ensure that the design pattern can be applied. Third, the process performs a parameterized transformation of the problem structure (i.e., the portion of the software design to be reorganized) into the target design. The parameters used in this step are the roles in a design pattern that must be filled by (e.g.) classes and operations. Fourth, the software engineer must address changes that the design pattern operator made to interfaces of existing classes. All classes that depend on these interfaces must be updated to be consistent with the changed design. Finally, the fifth step calls for the postconditions of the design pattern operator to be checked. The postconditions are difficult to formalize, and therefore they are described informally and may need further attention from the software engineer. The authors argue that recasting design patterns as operators, as opposed to building blocks, is favorable because the software engineer applying the design pattern only needs to know the parameters (i.e., the role-playing classes, operations, etc.) that will participate in the pattern. This removes the need to understand all of the details of implementing a design pattern from scratch.

Ó Cinnéide and Nixon [28] introduced the notion of decomposing design patterns (e.g., those of Gamma *et al*) into *minipatterns* (e.g., "encapsulate object construction" and "abstract access via an interface") and *minitransformations* (e.g., "replace class with interface"). They argue that by understanding these smaller elements, a developer can reason about how an existing model must change in order to incorporate a given design pattern while preserving the original behavior and semantics of the model. The authors later considered the automated application of minipatterns and minitransformations in order to produce better refactoring tools [27, 29]. How-

ever, they acknowledge that the approach can be automatically applied only under "green-field conditions", thus limiting its general applicability.

Albin-Amiot *et al.* [1] discuss the available tools and techniques for automating the use of design patterns with respect to building a new program or identifying design patterns in an existing program. They propose that the combination of a pattern description language, a set of pre-specified patterns, automated detection and identification of patterns, and automated source code refactoring is the correct approach to solving the larger problem of automating the detection, analysis, and application of design patterns. The authors present two prototype tools: `Patterns-Box`, which provides assistance to developers who are designing a new program, and `PTIDEJ`, which helps to identify design patterns in existing code. When used together, these prototypes can detect defects in existing software designs and assist in the correction of those defects by suggesting and automatically applying changes. Despite this increased focus on general automation, however, the approach does not consider the use of flexible search techniques such as evolutionary computation.

Jeon *et al.* [19] propose an automated approach to refactoring based on design patterns in Java programs. Their approach defines a set of inference rules for identifying portions of an existing design that could benefit from the introduction of a design pattern. Furthermore, the approach automatically identifies a refactoring strategy in order to introduce a given design pattern. The approach translates the target software design into set of PROLOG predicates, similar to the approach of Birkner [4] and the approach presented in this thesis. Each supported design pattern has a corresponding Prolog query that identifies classes that could fill the roles of the pattern. The authors present an example application of the `Abstract Factory` design pattern to a simple program.

Gomes *et al.* [13, 14] identified the need for automation of design pattern selection. They argue that existing approaches depend too heavily on human input in

order to choose which design patterns to apply and when to apply them. Using Case-Based Reasoning (CBR), a technique developed by the Artificial Intelligence community, they present a method that seeks to automate the selection and application process. By encoding the situation in which a design pattern was used in the past, a CBR system can learn to select and apply design patterns based on the structure of an existing program. Representing a design pattern in the form of a case (i.e., a chunk of experience), then, enables this approach to automate the selection and application of design patterns. In order to validate the approach, the authors constructed a prototype tool that consisted of a UML front-end and a CBR reasoning engine. Using a sample of 60 past design pattern applications and 25 sample class diagrams (containing three to five objects and no operations or attributes), the authors report that the prototype tool correctly identified 76% of the past design pattern applications as viable for the sample model being considered.

Our approach uses the key insight that Ó Cinnéide's minitransformations [27] can be understood as functions that transform one software design into another.

3.5 Summary

In this chapter, we reviewed the literature related to search-based refactoring, design pattern representation and detection, and automated instantiation of design patterns. Given the utility and ubiquity of refactorings, we recognize that automated refactoring has the potential to reduce significantly the amount of manual labor that software engineers must perform in order to maintain and extend existing software systems. However, existing automated refactoring approaches tend to couch the problem as one of pattern matching; i.e., first locate a sub-optimal portion of the design, and then make a pre-defined set of changes to that portion to improve it. This means that we must be able to specify all of the design flaws that we want to correct, which is difficult to achieve in general. By leveraging evolutionary compu-

tation and using software engineering metrics for guidance, search-based refactoring approaches have minimized the problem of identifying which portions of a design to refactor. However, these approaches have not addressed two key issues. First, we believe that the composition of multiple refactorings can generate richer, more complex sequences of refactoring steps. Following this line of thought, we believe that composition of multiple refactorings can result in the automated introduction of design patterns into existing software designs. This thesis addresses these issues through the use of genetic programming, the minitransformations of Ó Cinnéide, and software engineering metrics.

Chapter 4

Approach

In this chapter, we present our approach for the automated improvement of object-oriented software design through the introduction of design patterns.

4.1 Overview

Our approach to improve object-oriented software design is to use a genetic programming environment (a GP) that is guided by software engineering metrics to determine the optimal set of refactorings to apply to a software design. This approach has two objectives: first, to improve the quality of the design as measured by the QMOOD suite of object-oriented metrics; and second, to introduce design patterns when appropriate in order to improve the maintainability of the design. In this chapter, we describe the three major components of the GP: how individuals in the evolving population are represented, how those individuals are changed through mutation, and how newly-evolved individuals are evaluated. A data flow diagram that represents the approach is shown in Figure 4.1, and the remainder of this chapter describes the elements of the diagram.

[42]This page dedicated to the memory of Douglas Adams.

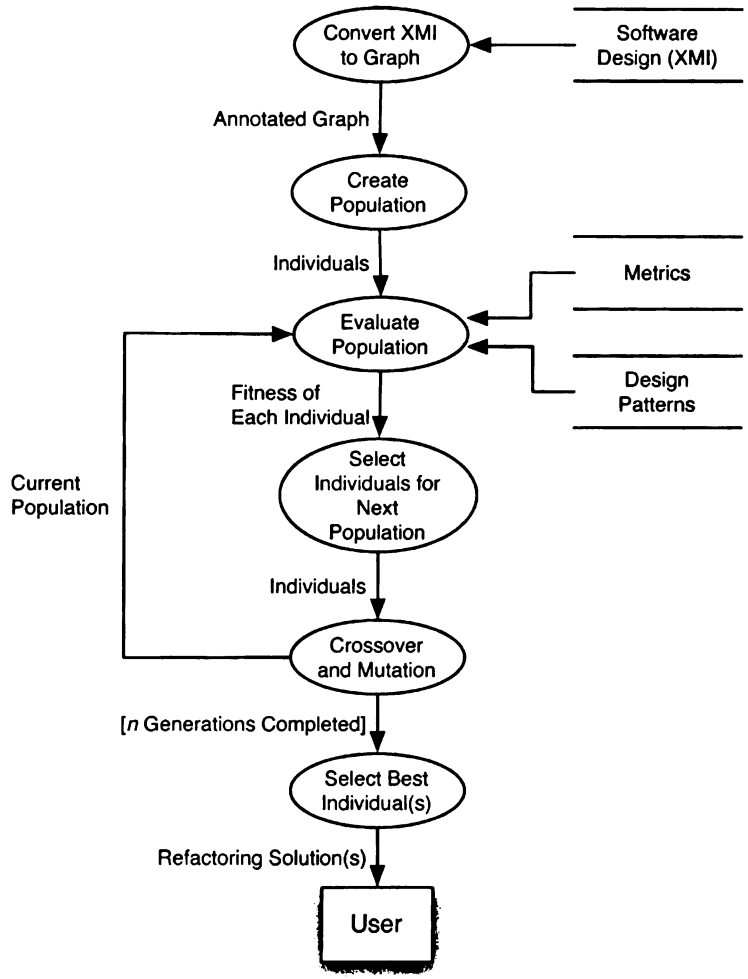


Figure 4.1: Data Flow Diagram of the Proposed Approach

4.2 Individual Representation

A GP contains a finite population of individuals that are given an optimization problem to solve. In this approach, we represent these individuals as a pair that includes a design graph and a transformation tree. A visual depiction of the relationships between a population, an individual, and the components of an individual is given in Figure 4.2.

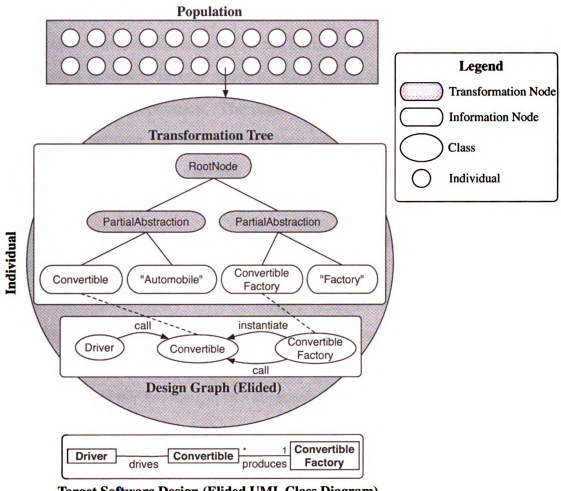


Figure 4.2: GP Components and Relationships

4.2.1 Design Graph

One of the inputs to our approach is the target software design, in the form of a UML class diagram, that we want to refactor to include design patterns. Our approach translates this class diagram into a graph (i.e., a set of vertices and a set of edges between those vertices) that is augmented with design meta-data, such as method invocations. We call this augmented graph structure a *design graph*. A simple UML class diagram is shown at the bottom of Figure 4.2, and the equivalent design graph is shown directly above it. Each vertex in a design graph represents a class, interface, or operation. The edges between these vertices denote semantic relationships between the entities that they represent. An edge has one of the following types: *aggregate*, *associate*, *compose*, *call*, *implement*, *inherit*, *instantiate*, or *own*. Each type represents a relationship between elements in an object-oriented design. Every individual in the population is given a copy of the design graph to modify.

Representing the target software design as a graph enables us to use common graph algorithms for analysis. Furthermore, a graph-based representation enables us to use metrics that are amenable to graph-based implementations, such as QMOOD [2]. To the best of our knowledge, this approach is the first to combine a graph-based representation of a software design with a GP to perform automated refactoring.

4.2.2 Transformation Tree

Each individual is responsible for modifying its copy of the design graph to improve its quality with respect to a set of metrics. The type of modifications to perform, and the order of those modifications, is decided by the individual's transformation tree. A sample transformation tree is shown in Figure 4.2. The transformation tree is a type of abstract syntax tree (AST). ASTs are typically used to represent the structure of a computer program, which comprises a series of code statements

that must be executed in a specific order. In this approach, the code statements are replaced by minitransformations that make well-defined changes to the design graph. These minitransformations are represented by nodes. When an individual is executed by the GP, the individual's transformation tree is executed in standard tree-traversal order. That is, the root node is executed first, and its children are executed recursively, beginning with the left-most node and proceeding from left to right. When a node is executed, it has the opportunity to analyze or modify the graph. A node can also take input from its child nodes, thus enabling multiple nodes to collaborate to make complex modifications to the graph. Next, we describe the types of nodes that appear in a transformation tree.

4.3 Design Change Mechanisms

We introduce two categories of transformation tree nodes in our approach. The first category, *transformation nodes*, comprises nodes that make changes to the target software design. These nodes are depicted as shaded ovals in the transformation tree shown in Figure 4.2. Transformation nodes require information from the target design to do their work. For example, a transformation node that represents the `PartialAbstraction` minitransformation needs to know the class from which it will construct an abstraction. This information is provided by the second category of nodes, *information nodes*, which represent elements in the target design (e.g., classes). These nodes are depicted as white ovals in the transformation tree in Figure 4.2. Similar to the order of parameters in a function call, the left-to-right order of the information nodes determines their role in the work of a transformation node. In the remainder of this section, we discuss each of these categories and the nodes that they comprise. We begin by describing a set of helper functions that perform simple tasks on behalf of the transformation nodes, such as creating an empty class in the target software design. Next, we discuss each of the transformation

nodes and each of the information nodes. We conclude with a discussion of the way in which the nodes are evaluated when a transformation tree is being executed.

4.3.1 Helper Functions

In this section we discuss each of the helper functions that supports the operation of the transformation nodes.

abstractClass

The `abstractAccess` function is declared as follows:

```
Interface abstractAccess(Class c, String name)
```

This function constructs and returns a new interface named *name* that contains all of the public methods of class *c*.

createEmptyClass

The `createEmptyClass` function is declared as follows:

```
Class createEmptyClass(String name)
```

This function constructs and returns an empty class named *name*.

createWrapperClass

The `createWrapperClass` function is declared as follows:

```
Class createWrapperClass(Interface iface, String wrapperName)
```

This function constructs and returns a new class called *wrapperName* that provides the same interface as the interface *iface* and delegates each method in the interface to a wrapped object of type *iface*.

makeAbstract

The `makeAbstract` function is declared as follows:

`Operation makeAbstract(Operation c, String newName)`

This function constructs and returns a method (operation) called *newName* that, when given the same arguments, will create the same object as the constructor operation *c*.

4.3.2 Transformation Nodes

Transformation nodes make well-defined changes (transformations) to the target software design. These nodes are adapted from Ó Cinnéide's minitransformations [27] and are designed to construct instances of the Gamma design patterns [12]. As demonstrated by Ó Cinnéide, the minitransformations are sufficiently powerful to construct instances of the Gamma design pattern subset that we support in this approach. The minitransformations were originally designed to introduce design patterns into program source code. In our approach, we adapted the minitransformations to perform design-level transformations, thus enabling their use in UML class diagrams. Next, we present a list of the transformation nodes that includes their inputs (provided by child nodes in the transformation tree), outputs, a graphical depiction of the transformation that is performed, and a pseudocode description of the transformation.

`AbstractAccess` Transformation

The inputs and output of the **`AbstractAccess`** transformation node are as follows:

Inputs: `class Context`, `class Concrete`, `interface Iface`

Output: `interface Iface`

The **`AbstractAccess`** transformation node modifies a class **`Context`** that directly uses another class **`Concrete`** so that **`Context`** accesses **`Concrete`** through an interface **`IConcrete`**. This change is illustrated in Figure 4.3. It is generally accepted that

such *decoupling*, or loosening of the relationships between two classes by utilizing an interface, facilitates the development of new classes and other design changes in order to meet future requirements [12]. This minitransformation, when used in combination with the `Abstraction` minitransformation, utilizes a newly-created interface in order to decouple one class from another.

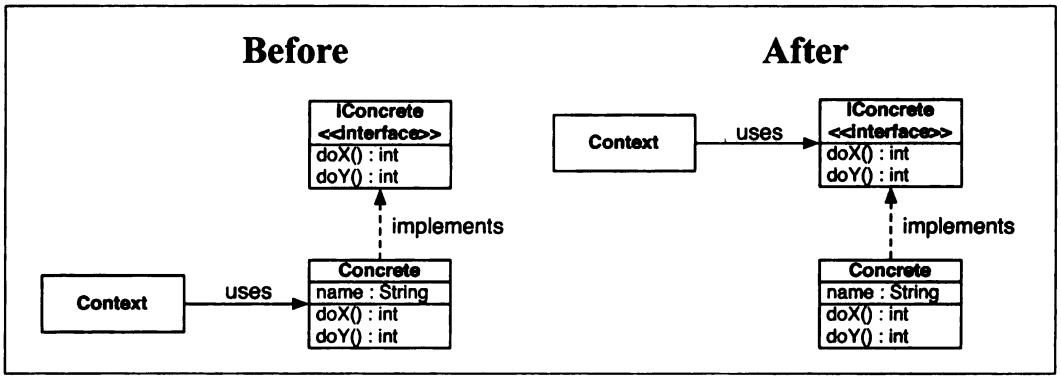


Figure 4.3: `Abstract Access` Minitransformation

Implementation. First, `AbstractAccess` node modifies the `Context` class to access the `concrete` class through the `Iface` interface. To accomplish this, it examines any outbound edges from `Context` that end at `Concrete`. For each such edge, the node creates a new edge from `Context` to `Iface` with the same type (e.g., `associate`). Finally, the `AbstractAccess` node returns `Iface` as its output. A pseudocode version of this algorithm is shown in Algorithm 1.

Algorithm 1 Pseudocode for `AbstractAccess` Transformation Node

```
AbstractAccess(graph G, class Ctx, class Conc, interface Iface)
1: edges = outEdges(Ctx)
2: for all e ∈ edges do
3:    if e.sink = Conc then
4:       G.addEdge(e.source, Iface, e.type)
5:       G.removeEdge(Ctx, e)
6:    end if
7: end for
8: return Iface
```

`Abstraction` Transformation

The inputs and output of the `Abstraction` transformation node are as follows:

Inputs: `class Product, string ifaceName`

Output: `interface Iface`

The `Abstraction` transformation node constructs a new interface that contains all of the public methods in an existing class, thus enabling other classes to take a more abstract view of the original class and any future classes that implement the interface. Introducing an interface in this fashion is a common refactoring step that restructures a software design to support additional functionality. For example, software that tracks vehicle inventory for an automotive dealership needs to track multiple vehicle types. These differing vehicle types will have many common characteristics, and a corresponding software design will include a class inheritance hierarchy that reflects those characteristics. When given a representative prototype, such as a class representing a mid-sized car, the `Abstraction` minitransformation can construct an interface for all classes that represent vehicles. A new vehicle-related class, then, only needs to implement the interface in order to integrate with the existing design.

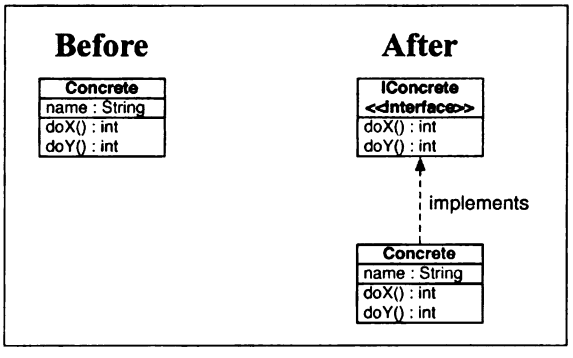


Figure 4.4: `Abstraction` Minitransformation

Implementation. The `Abstraction` transformation node creates a new interface `Iface`, whose name is derived from the string `ifaceName`, that has the same public methods and attributes as the `Product` class. Next, the node adds an `implements` edge between `Product` and the new interface. Finally, the node returns the new interface `Iface` as its output. A pseudocode version of this algorithm is shown in Algorithm 2.

Algorithm 2 Pseudocode for `Abstraction` Transformation Node

```
Abstraction(graph G, class Product, string IfaceName)
1: # Create an interface with the same public operations as Product
2: interface Iface = Helpers.abstractClass(Product, "I" + IfaceName)
3: G.addVertex(Iface)
4: G.addEdge(Product, Iface, IMPLEMENT)
5: return Iface
```

`Delegation` Transformation

The input and output of the `Delegation` transformation node are as follows:

Input: `class Owner`

Output: `class Delegate`

The `Delegation` transformation node is used to move part of an existing class to a component class, and to set up a delegation relationship from the existing class to its component. This change is illustrated in Figure 4.5. A class that has accumulated too many methods (sometimes called a "*god class*") may benefit from moving some of those methods to a separate *component* class. Since a method that is moved from one class to another will no longer have access to private members of the original class, those members must be made public. Thus, `Delegation` can require the public interface of a class to change.

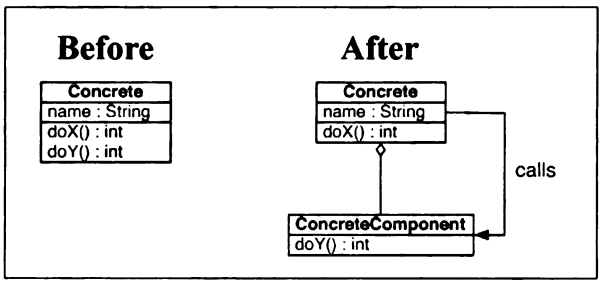


Figure 4.5: `Delegation` Minitransformation

Implementation. The `Delegation` transformation node creates a new empty class whose name is the concatenation of the original class (called `Owner` here) and the string "Delegate". For the purposes of this example, the class would be called `ownerComponent`. Next, two new edges are created. The edges are of type **aggregate** and `call`, respectively, starting at `Owner` and ending at the new `OwnerComponent` class. Now that the delegate class has been created, the transformation node delegates functionality by moving one or more of `Owner`'s operations to `OwnerComponent`. In our implementation, the transformation node randomly chooses two operations (if they are available) to delegate. Finally, the node returns `OwnerComponent` as its output. A pseudocode version of this algorithm is shown in Algorithm 3.

Algorithm 3 Pseudocode for `Delegation` Transformation Node

```
Delegation(graph G, class Owner)
1: string compName = Owner.name + "Component"
2: class Component = Helpers.createEmptyClass(compName)
3: G.addEdge(Owner, Component, AGGREGATE)
4: G.addEdge(Owner, Component, CALL)
5: # Choose two methods (if available) at random to delegate
6: Helpers.moveMethods(Owner, Component, 2)
7: return Component
```

`EncapsulateConstruction` Transformation

The inputs and output of the `EncapsulateConstruction` transformation node are as follows:

Inputs: `class Creator`, `class Product`

Output: `class Creator`

The `EncapsulateConstruction` transformation node weakens the binding between a class that creates instances of another class by relocating the code statements that perform instantiation into a dedicated method. This change is illustrated in Figure 4.6. In general, a class that contains many code statements that create objects is difficult to maintain. For example, if there is a change in the parameters of a class's constructor, then all statements that instantiate that class are invalidated and must be updated to use the modified parameters. However, we can resolve the problem by moving all object-creating code statements for a class `Shape` to a dedicated method `CreateShape` that is responsible for constructing and returning instances of `Shape`. By creating this dedicated method, we guarantee that any change to the constructor of class `Shape` only affects the code statements in a single method. The `EncapsulateConstruction` minitransformation constructs the dedicated method and modifies the relevant code statements to use the new method.

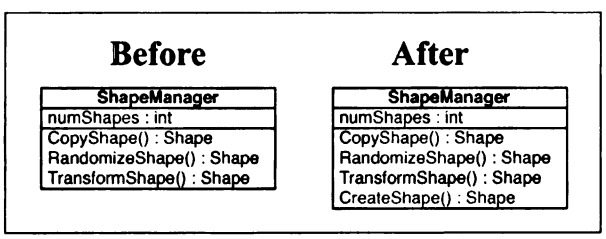


Figure 4.6: `EncapsulateConstruction` Minitransformation

Implementation. The `EncapsulateConstruction` transformation node creates a dedicated `createProduct` operation inside of the `Creator` class in order to localize the creation of `Product` instances to a specific location in the software design. To accomplish this, the node creates the `createProduct` operation and adds an `own` edge between the `Creator` class and the operation. Next, the node considers all

outgoing `instantiate` edges from the `Creator` class. If such an edge leads to the `Product` class, then the `instantiate` edge is removed, and a new `instantiate` edge is created between `createProduct` and `Product`. In addition, a `call` edge is created from `createProduct` to `Product`. Finally, the `Creator` class is returned as output. We note that the output from this transformation node is not a new design element. However, in order to satisfy the strong typing rules of the transformation tree, each transformation node must return an output. A pseudocode version of this algorithm is shown in Algorithm 4.

Algorithm 4 Pseudocode for `EncapsulateConstruction` Transformation Node

```
EncapsulateConstruction(graph G, class Creator, class Product)
 1: edges = outEdges(Creator)
 2: for all e ∈ edges do
 3:   if e.type = INSTANTIATE and e.sink = Product then
 4:     G.removeEdge(e)
 5:     string methodName = "create" + Product.name;
 6:     operation op = Helpers.makeAbstract(Creator, methodName)
 7:     G.addVertex(op)
 8:     G.addEdge(op, Product, INSTANTIATE)
 9:     G.addEdge(Creator, op, CALL)
10:   end if
11: end for
12: return Creator
```

`PartialAbstraction` Transformation

The inputs and output of the `PartialAbstraction` transformation node are as follows:

Inputs: `class Concrete`, `string NewAbstractName`

Output: `class Abstract`

The `PartialAbstraction` transformation node constructs a new abstract class from an existing (concrete) class and adds an `inherits` relationship from the concrete class to the abstract class. This change is illustrated in Figure 4.7. Growing the class inheritance hierarchy in a software design is a common task as the design matures and as requirements change over time. `PartialAbstraction` grows the inheritance hierarchy by creating a new abstract class that has the same methods as an existing class, thus enabling other classes to inherit the functionality of the existing class in a way that facilitates future maintenance.

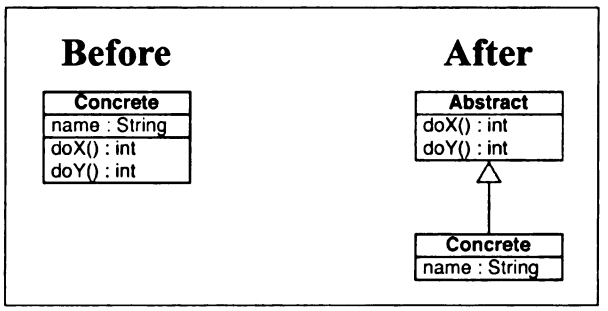


Figure 4.7: `PartialAbstraction` Minitransformation

Implementation. The `PartialAbstraction` transformation node creates a new abstract class whose name is derived from the string `NewAbstractName`. The node then creates an `inherit` relationship between the `Concrete` class and the new abstract class. Finally, the node returns the new abstract class as output. A pseudocode version of this algorithm is shown in Algorithm 5.

Algorithm 5 Pseudocode for `PartialAbstraction` Transformation Node

```
PartialAbstraction(graph G, class Concrete, string NewName)
1: class Abstract = Helpers.createEmptyClass("PAbstract"+NewName)
2: G.addVertex(Abstract)
3: G.addEdge(Concrete, Abstract, INHERIT)
4: return Abstract
```

`Wrapper` Transformation

The input and output of the `Wrapper` transformation node are as follows:

Input: `interface Iface`

Output: `interface WrapperIface`

The `Wrapper` transformation node wraps the functionality of an existing class with another class. This change is illustrated in Figure 4.8. Requests (e.g., method calls) to a wrapper object are forwarded to the wrapped object, and similarly the responses to those requests are passed back to the wrapper class and returned to the original calling object. This enables the wrapped class to be replaced at run time and loosens the coupling between the wrapped class and classes that use it.

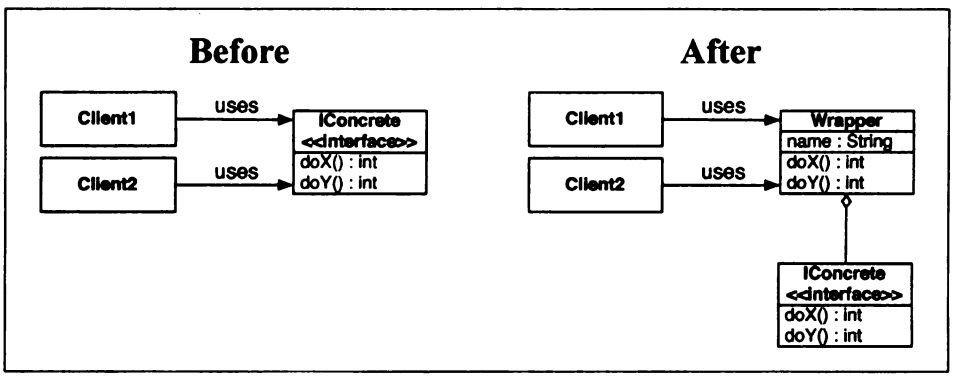


Figure 4.8: `Wrapper` Minitransformation

Implementation. The `Wrapper` transformation node wraps the functionality of a set of classes (i.e., classes that implement a given interface) inside of another class, thus enabling the wrapped class to be replaced at run-time. To accomplish this transformation, the node creates a new `IfaceWrapper` class and adds an **aggregate** edge between `IfaceWrapper` and the interface `Iface`. Semantically, this relationship can be understood as the aggregation of multiple *implementations* of the interface, and this facilitates run-time selection of the implementation that is most suitable.

Next, the node modifies the classes that use `Iface` so that they use `IfaceWrapper` instead. Ó Cinnéide's original `Wrapper` description specifies an additional input, namely a set of *client* classes that use the `Iface` interface, and specifies that only these classes should be modified to use `WrapperClass`. However, our implementation assumes that all classes that uses `Iface` should be modified to use `WrapperClass`. We argue that, generally, a designer would wrap all classes that implement an interface so that the specific class being used can be changed at run-time, and thus all implementation classes should be modified. A pseudocode version of this algorithm is shown in Algorithm 6.

Algorithm 6 Pseudocode for `Wrapper` Transformation Node

```
Wrapper(graph G, interface Iface)
 1: string wrapperName = Iface.name+"Wrapper"
 2: interface IfaceWrapper = Helpers.createWrapperClass(Iface, wrapperName,
    "wrapped" + Iface.name)
 3: edges = inEdges(Iface)
 4: for all e ∈ edges do
 5:    addEdge(e.source, IfaceWrapper, e.type)
 6:    removeEdge(e)
 7: end for
 8: G.addEdge(IfaceWrapper, Iface, AGGREGATE)
 9: G.addEdge(IfaceWrapper, Iface, CALL)
10: G.addEdge(IfaceWrapper, Iface, INSTANTIATE)
11: return IfaceWrapper
```

4.3.3 Information Nodes

Information nodes supply semantic information from the target software design to the transformation nodes. To use a natural language analogy, if transformation nodes are verbs, then information nodes are the nouns and adjectives that provide context for the verbs. For example, a transformation node that modifies a class will need to be informed of the specific class that it is to modify. We now discuss each of the information nodes in turn.

- `ClassNode` instances represent a class in the target design.
- `IfaceNode` instances represent an interface in the target design.
- `OperNode` instances represent an operation in the target design.
- `StringNode` instances represent a string that is derived from a dictionary of words that are relevant in the target design. This dictionary is typically generated from a requirements document.
- `RootNode` is a place-holder that has no effect on the target design and does not inform any transformation nodes. It is responsible for controlling the arity of the transformation tree; i.e., it loosely controls the number of transformation subtrees that evolve as the GP runs.

4.3.4 Lazy Node Evaluation

One of the key GP design challenges in this work was to ensure that new design elements (i.e., classes, interfaces, and operations) created during the execution of a transformation node are available for use by all other nodes in the same individual. For example, we assume that a `Wrapper` transformation node has created a new *wrapper* class named `MyWrapper`. We further assume that a `PartialAbstraction` transformation node is executed later and needs to use the `MyWrapper` class in order to build an abstract class with the same public interface. For this composition of transformation nodes to take place, the `Wrapper` transformation node must be a child of the `PartialAbstraction` node. However, this places a strong restriction on the types of composition among transformation nodes that can occur. Ideally, transformation nodes in *separate subtrees* can reuse one another's newly-created design elements. Furthermore, information nodes should be able to refer to classes, interfaces, and operations that did not exist in the original target design but were instead created by transformation nodes.

In order to provide a richer environment for composition of transformation nodes, we use a technique known as *lazy evaluation.* When a `ClassNode` is created during tree growing (e.g., during population initialization), it simply acts as a placeholder and does not refer to any specific class. When the tree containing the `ClassNode` is executed, the `ClassNode` will be visited, and at this point it will randomly choose a class from the *current* target design, which includes any classes that have been created by transformation nodes during tree execution. Once the `ClassNode` has chosen a class to represent, the choice is final. If the node becomes part of another individual's tree through subtree crossover, the chosen class is preserved. This technique ensures that new classes have a fair chance of being selected.

4.4 Evaluating Refactored Designs

When the GP evaluates an individual in the population, it executes the transformation tree associated with that individual. When this execution completes, the individual's copy of the target software design is in a modified state, and thus its quality must be re-evaluated. The GP uses this updated quality to determine whether the individual should be carried into the next generation of individuals. In this section, we discuss the factors that contribute to the quality of a modified target software design.

4.4.1 Design Quality

Once an individual has executed and modified its copy of the target software design, the quality of that modified design must be evaluated. In this approach, we evaluate its quality using the QMOOD metric suite. The modified software design is passed to QMOOD as input, and the metrics suite returns a real-valued number that represents the overall quality of the design, taking into account quality characteristics such

as readability and maintainability. This number is used as the *base fitness* for the design. Incentives and rewards (e.g., for individuals that evolve design pattern instances) can be added to the base fitness in order to encourage the evolution of solution characteristics that are appropriate for a given target software design.

4.4.2 Design Pattern Detection

The design change mechanisms (minitransformations) that are used in this approach are designed to create instances of the Gamma design patterns when they are composed. One of our primary objectives in this approach is to modify target software designs so that they incorporate design patterns. However, in order to identify candidate designs that include design patterns we must have a mechanism for detecting their presence. In this section, we describe the PROLOG-based design pattern detector that we developed for this detection task.

PROLOG [26] is a declarative logic programming language that supports queries over relations. To facilitate design pattern detection, we translate the target software design into relations that specify the classes and operations in the design, as well as the relationships between them (e.g., class inheritance or class-operation ownership). The relations for classes, interfaces, and operations are `cls`, `interface`, and `operation`, respectively. With this set of relations in hand, we construct one PROLOG query for each design pattern whose instances we wish to detect in a given software design. Our PROLOG queries are adapted from the work of Krämer *et al.* [21], which provided a query for each design pattern supported by our approach with the exception of `Abstract Factory`. A PROLOG query for the `Abstract Factory` design pattern is shown in Figure 4.9.

When an individual in the population is being evaluated, the PROLOG design pattern detector analyzes the individual's design graph for the presence of design

```
abstract_factory(AFact,CFact,AProd,CProd,Client) :-
  cls(AFact), cls(CFact), cls(AProd), cls(CProd),
  cls(Client), inherit(CFact,AFact),
  inherit(CProd,AProd), instantiate(CFact,CProd),
  fcall(Client,CProd), AFact \= CFact,
  AProd \= CProd, AFact \= Client,
  CFact \= Client, AProd \= Client,
  CProd \= Client, AFact \= AProd,
  AProd \= CFact.
```

Figure 4.9: PROLOG code for ABSTRACT FACTORY design pattern

pattern instances by running the query for each design pattern individually. When an instance is detected, Prolog returns the set of design elements that satisfy the query; i.e., the set of classes, interfaces, and operations that participate in the design pattern instance. In the case of the *Abstract Factory* design pattern, this set of elements must include classes that fill the following roles: *Abstract Factory* (AFact), *Concrete Factory* (CFact), *Abstract Product* (AProd), *Concrete Product* (CProd), and *Client* (Client), where '\=' is the negation of the boolean equality operator (i.e., *not equal to*). The GP records this set of elements for later processing.

4.4.3 Fitness Function

After the evaluation of an individual has completed, the GP assigns a fitness value, a real-valued number, to the individual. This value is the result of a mathematical function known as a *fitness function* that incorporates the QMOOD value for the target software design, the presence (or lack) of design pattern instances in that design, as well as factors related to the content of the individual's transformation tree. The fitness function that we developed for this approach is as follows:

Fitness = C_Q· QMOOD + C_R· patternReward - C_{NCP}· nodeCountPenalty +

C_{MSR}· matchingSequencesReward,

where C_Q, C_R, C_{NCP}, and C_{MSR} are coefficients that determine the relative weight of each term. These weights are parameters of the approach that may be tuned as appropriate for a given problem or domain. Next, we describe each term in the fitness function in detail.

- `QMOOD` provides a rough measure of overall design quality.

- **`patternReward`** adds a bonus if the target software design contains *at least one* design pattern instance. As a baseline, we have set this bonus at twice the QMOOD value. If the target software design does not contain a design pattern instance, then this term evaluates to zero.

- **`nodeCountPenalty`** provides a penalty that is proportionate to the number of transformation nodes in the transformation tree:[1] the penalty is intended to drive the GP to select individuals that have fewer transformation nodes, thus reducing the number of refactoring steps that are required to realize the individual's refactored software design. To compute this term, we multiply the QMOOD value by the number of transformation nodes in the transformation tree and multiply the result by a constant coefficient that determines the strength of the penalty.

- **`matchingSequencesReward`** assigns a bonus for specific sequences of transformation nodes. Ó Cinnéide specified sequences of minitransformations that, under green-field circumstances, would create instances of design patterns [27]. To drive individuals in the population to evolve design pattern instances, we assign a reward that is proportionate to the number of these sequences that are detected. (We note that other sequences of minitransformations are capable of creating design pattern instances, and we have observed this capability

[1]Some transformation nodes require more child nodes than others, and therefore we penalize in proportion to the number of transformation nodes rather than the total number of nodes in the transformation tree.

empirically. However, the GP is able to evolve design pattern instances more consistently when this reward is applied.) The GP maintains a record of the order in which transformation nodes are executed, and this record is consulted to determine when a reward should be assigned.

A note on minitransformation sequences

As discussed above, the `matchingSequencesReward` term in the fitness function awards a fitness bonus for individuals that apply specific sequences of minitransformations. The following is a list of these sequences. The design patterns whose instances are likely to result from the application of each sequence are given in parentheses.

- Wrapper, Abstraction, AbstractAccess (Adapter, Proxy)
- Wrapper (Bridge)
- Abstraction, AbstractAccess (Composite, Prototype)
- Abstraction, AbstractAccess, EncapsulateConstruction (Abstract Factory)
- Abstraction, AbstractAccess, Wrapper (Decorator)

4.5 Example

We now present a concrete example that describes the execution of a simple transformation tree and its corresponding effect on a target software design. Specifically, we demonstrate how an instance of the `Abstract Factory` design pattern is constructed by the GP. The execution is shown graphically in Figure 4.10. For this example, we assume that an individual is undergoing a fitness evaluation. A key component of that evaluation is executing the individual's transformation tree so that the GP

can determine whether the individual is capable of introducing a design pattern instance. The individual's transformation tree is shown at the top of Figure 4.10. The two shaded nodes in the transformation tree are transformation nodes, and therefore they are responsible for modifying the target software design. The nodes in the tree are executed in a post-traversal order. For example, `RootNode` executes its left child (i.e., $\texttt{PartialAbstraction}_1$) first, then its right child (i.e., $\texttt{PartialAbstraction}_2$), and finally itself. The child nodes perform the same execution recursively until the bottom of the tree is reached.

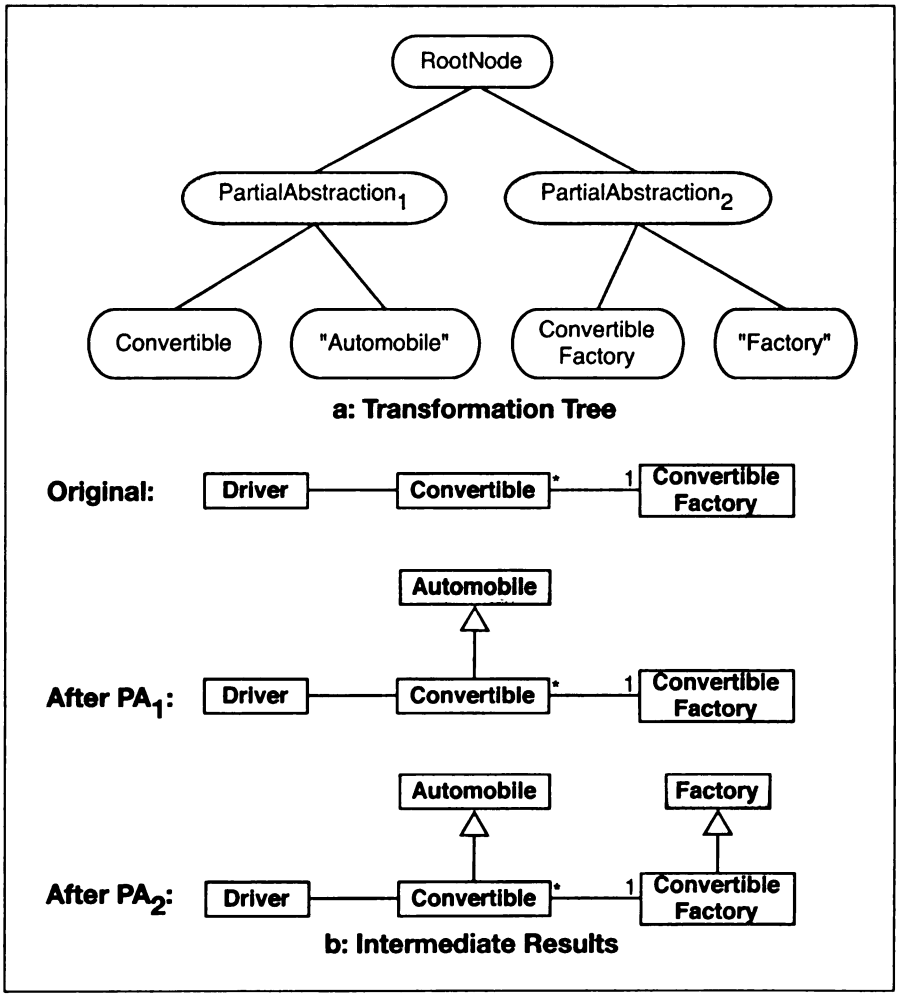


Figure 4.10: Example Transformation Tree Execution

In the lower half of the Figure 4.10 are three successive versions of an elided target software design that is being modified. These versions show the state of the

target software design at the beginning of the tree execution (denoted by "Original"), after the first `PartialAbstraction` node (denoted by "After PA_1"), and finally after the second `PartialAbstraction` node (denoted by "After PA_2"). When the first `PartialAbstraction` node is executed, a new superclass called **`Vehicle`** is introduced, and the existing `Convertible` class inherits from it. Finally, the remaining `PartialAbstraction` node executes, and another superclass called **`Factory`** is created, along with an inheritance relationship between **`ConvertibleFactory`** and **`Factory`**.

This simple example illustrates the construction of an **`Abstract Factory`** design pattern instance.

4.6 Summary

In this chapter, we presented a detailed discussion of our approach for the automated improvement of object-oriented software design through the introduction of design patterns.

Chapter 5

Validation

In this chapter, we validate our approach to automated software design improvement through a series of experiments. These experiments are conducted using a prototype implementation of the approach described in Chapter 4. First, we present an overview of the prototype and its run-time environment. Next, we discuss the experimental parameters and terms that are used throughout this chapter. Finally, we present the experiments and experimental results, along with a detailed discussion of the results.

5.1 Prototype Implementation

In this section, we describe the prototype implementation that we constructed to validate our approach, as well as the run-time environment for the experiments in the next sections. We implemented our prototype using ECJ[1] (Evolutionary Computation for Java), a Java-based framework that enables rapid prototyping of projects that use evolutionary computation to solve optimization problems. The following paragraphs describe the ways in which we extended ECJ to implement our approach.

[1]ECJ is available at `http://cs.gmu.edu/~eclab/projects/ecj/docs/`

5.1.1 Importing UML Class Diagrams

In order to improve the quality of a software design, we must first represent that design in a format that is easy to analyze and manipulate. During the modeling and design phase of software, its object-oriented structure is typically represented in a modeling language such as UML. In order to facilitate information exchange between modeling tools, the XML Interchange (XMI) format was developed. While there are several mutually-incompatible dialects of XMI implemented in tools such as Rational XDE and ArgoUML, the format provides a useful medium for importing model information from those tools.

For the approach proposed in this thesis, we use the XMI dialect implemented by ArgoUML [45]. To facilitate the use of this format, we developed a tool that translates the XMI data produced by ArgoUML into an graph-based structure that can be analyzed and manipulated by the GP.

5.1.2 Representing the Target Software Design as a Graph

Our approach represents the target software design (i.e., the design that we wish to refactor) as a graph. A graph is a set of vertices and a set of edges between those vertices. Our prototype uses the JGraph[2] library to construct graph instances. Since representing a software design as a graph requires us to annotate the vertices and edges with additional information, such as the type of relationship that exists between two classes, we extend the basic graph implementation in JGraph to include these details. We refer to this extended implementation as a *design graph*. Each vertex in a design graph has an associated type (e.g., class, interface, or operation). Similarly, each edge has an associated type (e.g., association or inheritance) that gives the relationship a semantic meaning. Each individual in the GP population is given a copy of the graph to modify.

[2]JGraph is available at `http://jgraph.com/`

In order to provide more design information for the GP and metrics to use, we make several assumptions about the relationships among classes and interfaces in the input model. For each association between two classes, in addition to creating an `associate` edge, we add two `call` edges, one going in each direction between the classes. If an operation in class `ClassA` returns an instance of a user-defined class `ClassB` (i.e., a class that is defined in the target software design), then we add an `instantiate` edge between `ClassA` and `ClassB` since it is likely that `ClassA` creates the instance that its operation returns. Similarly, for each class attribute whose type is a user-defined class, we add `instantiate` and `aggregate` edges between the class and the attribute.

5.1.3 Mutation Operators for Manipulating the Graph

The core of each individual in a GP population is its transformation tree. The transformation tree comprises transformation nodes and information nodes, and it is responsible for modifying the individual's design graph (and, therefore, the target software design). The specific changes that are made to the design graph depend on the type of nodes in the transformation tree and their position in the tree. The transformation nodes in this approach are implementations of the minitransformations proposed by Ó Cinnéide [27]. Since the minitransformations are defined as functions, they can be represented naturally as nodes in the tree-based representation used in genetic programming.

Each transformation node requires input to complete its work. This input is mostly provided by child *information* nodes, although other transformation nodes can provide input as well. Each information node is a reference to a specific element in the design graph. The information nodes represent classes, interfaces, and operations The child nodes are ordered from left to right, the role that each child node plays depends on its position in that order. For example, the `EncapsulateConstruction`

transformation node uses its left child node as the `Creator` class and its right child node as the `Product` class. In the language of genetic programming, these information nodes are *terminal nodes*,[3] since they do not have child nodes and thus represent the termination of a branch of the tree. Transformation nodes, conversely, are *non-terminal nodes* that have one or more child nodes.

Transformation Tree Node Typing Rules

The ECJ framework enforces strict typing rules on transformation tree nodes, thus guaranteeing that each transformation tree is valid by construction. This guarantee reduces the amount of processing overhead incurred by the GP, as it does not need to verify a transformation tree before executing it.

We now present a description of the typing rules that were developed for this approach. Our approach defines the following primitive node types: `ClassNode`, `IfaceNode`, `OperNode`, `StringNode`, `RootNode`, and `nil` (a dummy type). Each transformation tree in our approach has a single `RootNode` at its root. Both `ClassNode` and `IfaceNode` nodes can be children of the `RootNode`, and thus our approach defines a `ClassOrIfaceNode` set type that enables the `RootNode` to accept both types as children.

The following list specifies the child types that are accepted by each primitive node, along with the node's return type. The first six nodes correspond to the transformation nodes discussed in Chapter 4.

`AbstractAccess`

Child nodes:

1. `ClassNode`
2. `ClassNode`

[3]There is one exception: the `RootNode` is a non-terminal information node.

3. `IfaceNode`

Returns: `IfaceNode`

`Abstraction`

Child nodes:

1. `ClassNode`

2. `StringNode`

Returns: `IfaceNode`

`Delegation`

Child nodes:

1. `ClassNode`

Returns: `ClassNode`

`EncapsulateConstruction`

Child nodes:

1. `ClassNode`

2. `ClassNode`

Returns: `ClassNode`

`PartialAbstraction`

Child nodes:

1. `ClassNode`

2. `StringNode`

Returns: `ClassNode`

`Wrapper`

Child nodes:

1. `IfaceNode`

Returns: `ClassNode`

The remaining nodes correspond to information nodes.

`ClassNode`

Returns: `ClassNode`

`IfaceNode`

Returns: `IfaceNode`

`RootNode`

Child Nodes

1. `ClassOrIfaceNode`
2. `ClassOrIfaceNode`
3. `ClassOrIfaceNode`

Returns: `nil`

`StringNode`

Returns: `StringNode`

5.1.4 Implementing the QMOOD Metrics Suite

After an individual has modified the target software design, the quality of the modified design needs to be evaluated. Our approach begins this evaluation by applying the QMOOD metrics suite to the design. We implemented the individual QMOOD metrics as simple graph algorithms that analyze the structure of an design graph to discover the number of distinct class hierarchies in the target design, the average number of public methods across all classes, and so on. The software designs that are used in our initial experiments do not contain class attributes, and therefore our QMOOD computation does not consider the Cohesion Among Methods of a Class (CAM) metric. However, this is a limitation of the software designs and not of the approach. The values of the remaining ten metrics are used to compute a real-valued *base fitness* that represents the design's overall quality, taking into account quality characteristics that include readability, flexibility, and so on. In this implementation, each of these characteristics is given equal weight. However, the specific vector of weights that are assigned can be tuned for the application or domain in question.

5.1.5 Detecting Design Patterns with PROLOG and SQL

The second step of evaluating a modified target software design is to identify any new design pattern instances that have evolved in the design. To facilitate this detection step, we implemented two design pattern detectors. The first detector uses PROLOG, a logic programming language, and the second uses the Structured Query Language (SQL). The detector implementations are based on the open source Java-based packages JLogic[4] and HSQLDB[5], respectively. Both detectors require the target software design to be translated into a set of *facts*. In the context of our approach, facts are predicates that specify a property of the target software design,

[4]JLogic is available at `http://jlogic.sourceforge.net/`
[5]HSQLDB is available at `http://hsqldb.org/`

such as "`Automobile` is-a class". Once the target software design is represented as a collection of facts, the design pattern detectors can perform an exhaustive search for design pattern instances.

The design pattern detectors use *queries* (sometimes called *rules*) to search for facts that meet certain criteria. As a simple example, a query might return all classes that have an inheritance relationship with another class, an association relationship with yet another class, and contain more than two operations. Using these criteria, we specify queries that accurately identify design pattern instances. The PROLOG rules (queries) used in this thesis are adapted from Krämer and Prechelt [21], and the SQL queries are adapted from the work of Birkner [4]. An exhaustive list of these queries is given in Appendix A. Through experimentation, we observed that our SQL detector can scan a software design for design pattern instances more than 100 times faster than the PROLOG detector. However, despite producing equivalent results, we note that PROLOG queries are easier to read and more intuitive than the corresponding SQL queries, and therefore we use the PROLOG queries in this discussion.

5.1.6 Providing a Domain-Specific Vocabulary

During the course of their execution, several transformation nodes (e.g., `Wrapper`) introduce new classes, interfaces, or operations. These new design elements must have unique names, and therefore a mechanism is needed to provide names that are relevant to the design. Ensuring that a sensible name is chosen for a new design element is a nontrivial problem. For the approach proposed in this thesis, we assume that a requirements document is available for the target software design. We then generate a list of nouns from the requirements document and use it as an additional input for the GP.

There are two cases when a name is required for a new design element. First, when a new `StringNode` is created, it selects a random noun from the list to use as its name. This value is then passed to the `StringNode`'s parent node, which is typically a transformation node. Second, several of the transformation nodes create new design elements that are closely related to existing elements. For example, a `Delegation` node might create a `DrawingComponent` class as a container for delegating some functionality from an existing `Drawing` class. Since the new component class (`DrawingComponent`) is tightly coupled to the `Drawing` class, we designed the `Delegation` transformation node to reuse its name. In this scenario, the transformation node does not need to select a new name from the list of nouns. If the chosen name already exists in the design, then a numerical suffix is appended to create a unique name (e.g., `DrawingComponent`) might become `DrawingComponent2`.

While this solution ensures that unique string-based names can be assigned to new design elements, we note that the solution is ad-hoc in nature and requires developer review to ensure that the chosen noun is sensible.

5.2 Run-time Environment and Parameters

The experiments presented in this chapter were run on a 128-node, 1024-core cluster of 2.3GHz machines with 8GB of memory running SuSE Linux Enterprise Server 10 (TM) with kernel version 2.6.16. For each model, we conducted five independent runs using random seed values in the range from 1 to 5. We refer to each of these runs as a *trial*. The random seeds are used by the Java random number generator and are instrumental in generating the initial population of random individuals, as well as determining the subset of those individuals that will be mutated. Performing multiple trials helps to control for edge cases that are not representative of an average run of the GP. Only the best individual in a given trial is considered in our results.

For all of the experiments in this paper, the mutation and recombination of individuals proceeds as follows. When individuals are selected to be placed into the next generation, 90% of those individuals will be mutated and then recombined with another individual. The remaining 10% of selected individuals will be placed into the next generation unchanged. These probabilities are commonly used in genetic programming-based approaches [20].

5.3 Hypotheses

For each of the experiments discussed in this chapter, we present two hypotheses: the *null* hypothesis takes a *no-change* perspective, stating that the proposed approach will have no discernible effect on the software design being refactored (e.g., in terms of the overall design quality or number of design pattern instances). The *alternative* hypothesis, conversely, states that the approach, when applied to a software design, has a discernible and significant effect on the design. The experiments that follow are designed to test the proposed approach in a way that will enable us to reject one hypothesis and accept the other.

5.4 Experiments

In order to evaluate the proposed approach and explore the effect of various parameter values on its ability to evolve a diverse set of design pattern instances in software designs, we developed a series of five experiments. The first four experiments validate the approach on a set of small UML models, most of which contain 10 to 15 classes. These models were originally used for a study of using case-based reasoning to automatically apply design patterns [13]. For example, one model represents the high-level workings of an accounting office, while another represents the design of a computer game. The set comprises 58 models in total, and each model represents a

software system that solves a unique problem. We document our process of selecting appropriate GP parameters (e.g., rewards and penalties) by describing the experiments that we conducted to compare the effects of various parameter values. The fifth experiment validates the approach on a larger UML model of a web-based repository of software models. The input models (software designs) for these experiments contain only UML class diagrams.

In experiments I-IV, we ran the GP for 100 generations with a population size of 100 individuals. The tournament size was 7 individuals. These parameters are typical for a genetic programming experiment and tend to provide enough running time and population diversity for a GP to discover high-quality solutions [37].

5.4.1 Experiment I: No Rewards or Penalties (Baseline)

The objective of this first experiment set is to evaluate our approach on a diverse set of models that were constructed without the use of design patterns, thus assessing its ability to introduce design pattern instances in such a scenario. For each of the 58 models, we performed the following experiment.

Setup and Parameters. In this experiment, the coefficients for the `patternReward`, `nodeCountPenalty`, and `matchingSequencesReward` terms in the fitness function were 0.0 in order to evaluate the efficacy of the implementation without any rewards or penalties.

Null Hypothesis (Base Case): *There is no change with respect to the number of design pattern instances in each trial when the GP terminates; i.e., none of the trials contains any new design pattern instances.*

Alternative Hypothesis: *A majority of the models contain a newly-evolved design pattern instance in at least one trial when the GP terminates.*

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.1. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 5.94. There were 3103 new design pattern instances that evolved during the run, and 40 of the 58 (69%) models evolved at least one new design pattern instance. This is an average of 10.7 newly-evolved design pattern instances per trial. From these initial results, it is clear that the approach is capable of constructing new design pattern instances in existing software designs.

Design Pattern Name	**Number of Instances**
Abstract Factory	1
Adapter	176
Bridge	7
Composite	24
Decorator	66
Prototype	2758
Proxy	71
Total	**3103**

Table 5.1: Experiment I: Evolved Design Pattern Instances

The trials in this experiment set ran for five minutes on average. The mean number of classes in the models was 10.9 at the beginning of the run and 77.1 at the end, showing a strong increase due to the nature of the minitransformations and the introduction of design patterns. Our detector found an average of 9.8 design pattern instances at the start of the run. Of the 290 trials (58 models and five seeds per model), there were 166 trials that did not evolve a design pattern instance. However, 40 of the 58 models evolved at least one new design pattern instance. Therefore, we reject the null hypothesis and accept the alternative hypothesis. In the remaining

experiments, we explore various options for increasing the frequency of evolution of design pattern instances.

5.4.2 Experiment II: Penalizing Large GP Trees

One of the design goals for the proposed approach is to generate sequences of refactoring steps of a *reasonable size* for a software engineer to understand and, if automation is not desired, to apply manually. Each individual in a GP population contains a transformation tree that, in turn, contains transformation nodes that modify the target software design. Roughly speaking, the number of refactoring steps represented by an individual's transformation tree is equal to the number of transformation nodes (minitransformations) in the tree. Thus, in order to encourage a population of individuals to generate shorter sequences of refactoring steps, in this experiment set we assign each individual a penalty proportionate to the number of transformation nodes. For all treatments in this experiment, this penalty is the product of a coefficient (e.g., 0.0025), the number of transformation nodes in the individual's transformation tree, and the QMOOD quality of the individual. The intent of this penalty is to apply a small, persistent pressure on individuals to evolve transformation tree that contain fewer transformation nodes.

Null Hypothesis (Base Case): *There is no decrease in the number of transformation nodes when individuals are given a fitness penalty proportionate to the number of transformation nodes, as compared with Experiment I.*

Alternative Hypothesis: *There is a decrease in the number of transformation nodes when individuals are given a fitness penalty proportionate to the number of transformation nodes, as compared with Experiment I.*

First Treatment: 0.0025 Coefficient

Setup and Parameters. In this first treatment, we choose the value 0.0025 for the coefficient of the `nodeCountPenalty` term in the fitness function. While the resulting penalty is small, we believe that it will provide a persistent and effective pressure toward trees with fewer transformation nodes. The coefficients for the `patternReward` and `matchingSequencesReward` terms are both set to 0.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.2. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 6.87, a slight increase in comparison to the control, Experiment I, and the average fitness of those best individuals was 3.40, a sharp decrease.

Design Pattern Name	**Number of Instances**
Abstract Factory	1
Adapter	239
Bridge	9
Composite	45
Decorator	20
Prototype	2681
Proxy	133
Total	**3128**

Table 5.2: Experiment II, First Treatment: Evolved Design Pattern Instances

Second Treatment: 0.025 Coefficient

Setup and Parameters. In this treatment, we choose the value 0.025 for the coefficient of the `nodeCountPenalty` term in the fitness function. The coefficients for the `patternReward` and `matchingSequencesReward` terms are both set to 0.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.3. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 5.86, and the average fitness of those best individuals was 2.57. Each model contained at least one new design pattern instance at the end of the run. In addition, the mean number of transformation nodes decreased slightly with respect to the control (5.94 in Experiment I).

Design Pattern Name	**Number of Instances**
Abstract Factory	4
Adapter	166
Bridge	5
Composite	10
Decorator	36
Prototype	1845
Proxy	73
Total	**2139**

Table 5.3: Experiment II, Second Treatment: Evolved Design Pattern Instances

Third Treatment: 0.25 Coefficient

Setup and Parameters. In this treatment, we choose the value 0.25 for the coefficient of the `nodeCountPenalty` term in the fitness function. The coefficients for the `patternReward` and `matchingSequencesReward` terms are both set to 0.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.4. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 0.67, and the average fitness of those best individuals was 1.05, both of which are a sharp decrease from the control (Experiment I).

Design Pattern Name	Number of Instances
Abstract Factory	9
Adapter	38
Bridge	0
Composite	0
Decorator	36
Prototype	198
Proxy	134
Total	**415**

Table 5.4: Experiment II, Third Treatment: Evolved Design Pattern Instances

Discussion

In this experiment, we demonstrated that applying a penalty in proportion to the number of transformation nodes in an individual's transformation tree is an effective mechanism for limiting the growth of the tree. There was no observed difference in the running time of these trials compared with those in Experiment I. The average

number of transformation nodes across all of the models for each treatment is given in Table 5.5. For comparison, the average number of transformation nodes from Experiment I is given as a control and is shown in the table as treatment C. While the greatest decrease was observed in Treatment 3, we believe that this penalty is too strong and will not provide enough flexibility for the GP to discover useful solutions. Therefore, we choose the smaller coefficient value of 0.025 for C_{NCP}. By keeping the number of transformation nodes at a reasonable level, we ensure that the sequences of refactorings generated by our approach can reasonably be applied by hand.

Treatment	C_{NCP}	**# of Trans. Nodes**
C	0.0	5.94
1	0.0025	6.87
2	0.025	5.86
3	0.25	0.67

Table 5.5: Experiment II: Number of Transformation Nodes

Now that we have examined the effect of a penalty for longer sequences of refactoring steps, the remaining experiments in this chapter will use the value 0.025 for the `nodeCountPenalty` term in the GP fitness function. Using this coefficient in the fitness function resulted in transformation trees that had an average of 5.86 transformation nodes. We believe that this number of design transformations is reasonable for a designer to understand and apply by hand, and it is large enough to facilitate the construction of design pattern instances.

5.4.3 Experiment III: Rewarding Evolution of Design Pattern Instances

In this section, we turn to the problem of how to encourage individuals to evolve a greater number of design pattern instances. We examine this problem by introducing a new reward that is the product of a real-valued coefficient and the QMOOD quality of an individual. Effectively, this reward improves an individual's fitness by a fixed percentage that is equal to the value of the coefficient. For example, a coefficient of 0.2 provides a fitness boost that is equal to 20% of the individual's base fitness. We examine several values for the coefficient in this experiment set that comprises four treatments.

Null Hypothesis (Base Case): *There is no increase in the number of design pattern instances when individuals are given a fitness bonus for design pattern presence, as compared with Treatment 2 from Experiment II.*

Alternative Hypothesis: *There is an increase in the number of design pattern instances when individuals are given a fitness bonus for design pattern presence, as compared with Treatment 2 from Experiment II.*

First Treatment: 0.125 coefficient

Setup and Parameters. In this first treatment, we choose the value 0.125 for the coefficient of the `patternReward` term in the fitness function. The coefficient for the `nodePenalty` term is 0.025, and the coefficient for the `matchingSequencesReward` term is 0.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.6. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 6.42. Of

the 58 models, 2 showed a statistically significant increase in the number of design patterns compared with Treatment 2 from Experiment II.

Design Pattern Name	**Number of Instances**
Abstract Factory	5
Adapter	248
Bridge	14
Composite	44
Decorator	57
Prototype	3546
Proxy	109
Total	**4023**

Table 5.6: Experiment III, First Treatment: Evolved Design Pattern Instances

Second Treatment: 0.25 Coefficient

Setup and Parameters. In this treatment, we choose the value 0.25 for the coefficient of the `patternReward` term in the fitness function. The coefficient for the `nodePenalty` term is 0.025, and the coefficient for the `matchingSequencesReward` term is 0.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.7. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 6.66. Of the 58 models, 4 showed a statistically significant increase in the number of design patterns compared with Treatment 2 from Experiment II.

Design Pattern Name	Number of Instances
Abstract Factory	9
Adapter	289
Bridge	4
Composite	37
Decorator	86
Prototype	2979
Proxy	158
Total	**3562**

Table 5.7: Experiment III, Second Treatment: Evolved Design Pattern Instances

Third Treatment: 0.5 Coefficient

Setup and Parameters. In this treatment, we choose the value 0.5 for the coefficient of the `patternReward` term in the fitness function. The coefficient for the `nodePenalty` term is 0.025, and the coefficient for the `matchingSequencesReward` term is 0.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.8. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 6.68. Of the 58 models, 7 showed a statistically significant increase in the number of design patterns compared with Treatment 2 from Experiment II.

Fourth Treatment: 1.0 Coefficient

Setup and Parameters. In this treatment, we choose the value 1.0 for the coefficient of the `patternReward` term in the fitness function. The coefficient for the

Design Pattern Name	Number of Instances
Abstract Factory	4
Adapter	365
Bridge	2
Composite	29
Decorator	31
Prototype	3080
Proxy	174
Total	**3685**

Table 5.8: Experiment III, Third Treatment: Evolved Design Pattern Instances

`nodePenalty` term is 0.025, and the coefficient for the `matchingSequencesReward` term is 0.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.9. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 6.79. Of the 58 models, 17 showed a statistically significant increase in the number of design patterns compared with Treatment 2 from Experiment II.

Fifth Treatment: 2.0 Coefficient

Setup and Parameters. In this treatment, we choose the value 2.0 for the coefficient of the `patternReward` term in the fitness function. The coefficient for the `nodePenalty` term is 0.0025, and the coefficient for the `matchingSequencesReward` term is 0.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.10. The mean number of transformation nodes

Design Pattern Name	Number of Instances
Abstract Factory	8
Adapter	360
Bridge	15
Composite	47
Decorator	89
Prototype	4018
Proxy	259
Total	**4796**

Table 5.9: Experiment III, Fourth Treatment: Evolved Design Pattern Instances

in the best individuals' transformation trees at the end of the run was 5.85. Of the 58 models, 9 showed a statistically significant increase in the number of design patterns compared with Treatment 2 from Experiment II.

Design Pattern Name	Number of Instances
Abstract Factory	14
Adapter	400
Bridge	12
Composite	75
Decorator	63
Prototype	3454
Proxy	247
Total	**4265**

Table 5.10: Experiment III, Fifth Treatment: Evolved Design Pattern Instances

Discussion

In this experiment, we considered several values for the coefficient of the **patternReward** term in the fitness function. There was no observed difference in the running time of these trials compared with those in Experiments I and II. While increasing the **patternReward** coefficient did not result in a consistently higher number of design pattern instances, it is clear from the results that there is a significant increasing trend in the number of instances, and more design pattern instances evolved in comparison to Experiment II. Therefore, we reject the null hypothesis and accept the alternative hypothesis. The number of models showing a statistically significant increase ($p < 0.05$) in the number of design pattern instances is shown for each treatment in Table 5.11. Based on the results, we observe that the **patternReward** coefficient value 1.0 produces the best combination of design pattern diversity and average number of new design patterns instances, and a greater number of models showed a statistically significantly higher number of evolved design pattern instances when using the coefficient value 1.0. We use this coefficient value for the remaining experiments.

Treatment	C_{PR}	**# of Models**
1	0.125	2
2	0.25	4
3	0.5	7
4	1.0	17
5	2.0	9

Table 5.11: Experiment III: Models with Significant Increase in DP Instances

5.4.4 Experiment IV: Rewarding Specific Sequences of Mini-transformations

Now that we have considered the coefficients for the `nodeCountPenalty` and `patternReward` terms in the fitness function, we turn our attention to the final term: `matchingSequencesReward`. This reward is given to individuals whose transformation trees contain transformation nodes in specific sequences that are known to produce design pattern instances, as discussed in Section 4.4.3. The objective of this reward is to encourage individuals to use these sequences. However, the reward must be small enough that individuals can explore other sequences.

Null Hypothesis (Base Case): *There is no increase in the number of design pattern instances when individuals are given a fitness bonus for using specific sequences of transformation nodes, as compared with Treatment 4 of Experiment III.*

Alternative Hypothesis: *There is an increase in the number of design pattern instances when individuals are given a fitness bonus for using specific sequences of transformation nodes, as compared with Treatment 4 of Experiment III.*

First Treatment: 0.025 Coefficient

Setup and Parameters. In this treatment, we choose the value 0.025 for the coefficient of the `matchingSequencesReward` term in the fitness function. The coefficient for the `nodePenalty` term is 0.025, and the coefficient for the `patternReward` term is 1.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.12. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 6.87. Of

the 58 models, 1 showed a statistically significant increase in the number of design patterns compared with Treatment 4 from Experiment III.

Design Pattern Name	Number of Instances
Abstract Factory	6
Adapter	315
Bridge	13
Composite	88
Decorator	105
Prototype	4026
Proxy	219
Total	**4772**

Table 5.12: Experiment IV, First Treatment: Evolved Design Pattern Instances

Second Treatment: 0.25 Coefficient

Setup and Parameters. In this treatment, we choose the value 0.25 for the coefficient of the `matchingSequencesReward` term in the fitness function. The coefficient for the `nodePenalty` term is 0.025, and the coefficient for the `patternReward` term is 1.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.13. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 6.86. Of the 58 models, 2 showed a statistically significant increase in the number of design patterns compared with Treatment 4 from Experiment III.

Design Pattern Name	Number of Instances
Abstract Factory	18
Adapter	423
Bridge	6
Composite	85
Decorator	74
Prototype	4291
Proxy	267
Total	**5164**

Table 5.13: Experiment IV, Second Treatment: Evolved Design Pattern Instances

Third Treatment: 0.5 Coefficient

Setup and Parameters. In this treatment, we choose the value 0.5 for the coefficient of the `matchingSequencesReward` term in the fitness function. The coefficient for the `nodePenalty` term is 0.025, and the coefficient for the `patternReward` term is 1.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.14. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 6.90. Of the 58 models, 3 showed a statistically significant increase in the number of design patterns compared with Treatment 4 from Experiment III.

Fourth Treatment: 1.0 Coefficient

Setup and Parameters. In this treatment, we choose the value 1.0 for the coefficient of the `matchingSequencesReward` term in the fitness function. The coefficient

Design Pattern Name	Number of Instances
Abstract Factory	5
Adapter	288
Bridge	1
Composite	55
Decorator	27
Prototype	4445
Proxy	251
Total	**5072**

Table 5.14: Experiment IV, Third Treatment: Evolved Design Pattern Instances

for the `nodePenalty` term is 0.025, and the coefficient for the `patternReward` term is 1.0.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.15. The mean number of transformation nodes in the best individuals' transformation trees at the end of the run was 7.23. Of the 58 models, 4 showed a statistically significant increase in the number of design patterns compared with Treatment 4 from Experiment III.

Fifth Treatment: 2.0 Coefficient

Setup and Parameters. In this treatment, we choose the value 2.0 for the coefficient of the `matchingSequencesReward` term in the fitness function. The coefficient for the `nodePenalty` term is 0.0025, and the coefficient for the `patternReward` term is 0.5.

Results. The number and type of design pattern instances that evolved during this experiment are given in Table 5.16. The mean number of transformation nodes

Design Pattern Name	Number of Instances
Abstract Factory	19
Adapter	447
Bridge	11
Composite	54
Decorator	92
Prototype	4127
Proxy	346
Total	**5096**

Table 5.15: Experiment IV, Fourth Treatment: Evolved Design Pattern Instances

in the best individuals' transformation trees at the end of the run was 7.14. Of the 58 models, 2 showed a statistically significant increase in the number of design patterns compared with Treatment 4 from Experiment III.

Design Pattern Name	Number of Instances
Abstract Factory	21
Adapter	400
Bridge	25
Composite	99
Decorator	83
Prototype	3541
Proxy	343
Total	**4512**

Table 5.16: Experiment IV, Fifth Treatment: Evolved Design Pattern Instances

Discussion

In this experiment, we considered several values for the coefficient of the `matchingSequencesReward` term in the fitness function. There was no observed difference in the running time of these trials compared with those in Experiments I-III. Based on the results, we observe a steady increase in the number of evolved design pattern instances when the reward is increased. Therefore, we reject the null hypothesis and accept the alternative hypothesis. The number of models showing a statistically significant ($p < 0.05$) increase in the number of design patterns is shown for each treatment in Table 5.17. The largest number of models showed a statistically significant increase in Treatment 4. We choose the value 1.0 as the `matchingSequencesReward` coefficient for the remaining experiment.

Treatment	C_{MSR}	**# of Models**
1	0.125	1
2	0.25	2
3	0.5	3
4	1.0	4
5	2.0	2

Table 5.17: Experiment IV: Models with Significant Increase in DP Instances

5.4.5 Experiment V: REMODD Case Study

This experiment set applies the approach to a model of the Repository for Model-Driven Development, or REMODD [11]. REMODD is a web-based repository for model-driven engineering artifacts, including models, test cases, documentation, code, etc. The design for REMODD contains 23 classes. This experiment is designed to test the approach on a medium-sized software design that already contains design pattern instances.

Null Hypothesis (Base Case): *There is no change with respect to the number of design pattern instances in this design; i.e., none of the trials contains a design pattern instance.*

Alternative Hypothesis: *At least one trial contains a newly-evolved design pattern instance when the GP terminates.*

Setup and Parameters. In this experiment, we ran the GP for 25 generations with a population size of 100 individuals. The tournament size was 7 individuals. Drawing from the conclusions of the previous experiments, the coefficients for the `patternReward`, `nodeCountPenalty`, and `matchingSequencesReward` terms in the fitness function were 0.5, 0.025, and 0.5, respectively.

Results. The trials for this experiment took an average of six days to complete. The longer running time is largely due to the increased complexity (i.e., a greater number of classes and associations between them) of the ReMoDD design compared with the designs in the first experimental set. This increased complexity requires the design pattern detector to search a much larger space for possible design patterns, and the running time scales accordingly. The number of classes in the model was 23 at the beginning of the run, and the mean number of classes at the end was 38.8. This is a more significant increase than we saw in the first experimental set. However, this software design contains twice as many classes as an average design in the first experimental set, and this creates a larger set of inputs for the minitransformations. As a result, we expect to see a larger number of design pattern instances evolving in each trial, and this tends to require a greater number of classes. Our detector found 136 candidate design pattern instances in the original design. Table 5.18 depicts the number and type of design pattern instances that evolved during the experiment. The mean number of transformation nodes in the best individuals was 4.50. There

were a total of 61 new design pattern instances that evolved during the run. This is an average of 12.20 design pattern instances per trial. Each of the five trials evolved at least one design pattern instance. Based on these results, at least one trial evolved a new design pattern instance. Therefore, we reject the null hypothesis and accept the alternative hypothesis.

Design Pattern Name	**Number of Instances**
Abstract Factory	2
Adapter	13
Bridge	3
Composite	0
Decorator	2
Prototype	26
Proxy	15
Total	**61**

Table 5.18: Experiment V: Evolved Design Pattern Instances

5.5 Discussion

The contrast among trials with respect to the number of evolved design pattern instances reflects the role of randomness in the initial population of individuals, as well as the multi-dimensional nature of the solution space. The trials that did not evolve new design pattern instances consistently produced solutions with a higher QMOOD value, and subsequently a higher fitness value, than other trials for the same model. This suggests that despite improvements in quality that often accompany the introduction of design patterns, their introduction is not always the best course of action when one is trying to improve design quality. However, it must be

noted that these results only use one quality model (i.e., QMOOD) and therefore may not be representative of all quality models.

We observe that specific *types* of design patterns are either over- or under-represented in these results. For example, the `Prototype` design pattern appears very often in experiments I-IV, and it is also the most prolific design pattern type in the final experiment. This distribution is due to the differing complexity of the design patterns. For example, an instance the *Bridge* design pattern comprises more classes and associations than an instance of the *Adapter* design pattern, and this increases the likelihood that an instance of *Adapter* will evolve.

The results of these experiments are encouraging. They show that the proposed approach scales to medium-sized software designs. Furthermore, the variety of design pattern types that evolved provides the human developer multiple options for improving the target software design, and each evolved design pattern instance provides a step-by-step procedure for implementing the instance, thus reducing the manual design (or redesign) labor that the developer must endure.

5.6 Summary

In this chapter, we presented experimental validation for the approach that is proposed in this thesis. First, we discussed the prototype implementation that was developed to validate the approach. Next, we discussed the experimental parameters, hypotheses, and terms that were used throughout the chapter. We then presented the series of experiments that were conducted to explore the parameter spaces and to identify the most suitable parameter values. Finally, we presented the results of a real-world case study and discussed the consequences of the experimental results.

Chapter 6

Conclusions and Future Work

6.1 Summary

In this thesis, we presented an approach for automated improvement of software design that leverages evolutionary computation, software engineering metrics, and the introduction of design patterns. Specifically, we proposed the use of Ó Cinnéide's minitransformations as mutation operators in a GP and the use of software engineering metrics to guide the process of selecting individuals to survive into subsequent generations. The minitransformations were originally discovered as part of a search for *mini-patterns*, or design patterns within design patterns [27]. The mini-patterns, it was hoped, would enable us to develop better automated refactoring techniques that could, for example, automatically introduce design patterns into a design that was constructed without them. By recognizing that these minitransformations are function-like constructs that can be composed to generate novel refactorings, we were able to construct a genetic programming-based approach for deciding the best sequence of refactorings to apply to an existing design.

Through a set of five experiments, we showed that our approach is capable of simultaneously improving the quality of a software design (as measured by software engineering metrics) and automatically introducing design pattern instances, a com-

bination that has not previously been considered. First, we demonstrated that the approach is feasible and can construct instances of design patterns in existing designs with guidance from the QMOOD metrics suite only. Next, we identified three optimizations that can be used to control the number and type of design patterns that evolve. One of our design goals was to ensure that a human software engineer could understand the entire set of refactoring steps that our approach recommended. Therefore, the first optimization penalizes individuals in proportion to the number of transformation nodes (minitransformations) in their transformation trees, thus reducing the number of suggested refactoring steps on average. We demonstrated the effectiveness of this optimization in the second experiment. The second optimization introduced a reward for individuals that evolved at least one design pattern instance. The amount of reward was a fraction of the QMOOD quality value for the design, and we demonstrated in the third experiment that this reward tends to increase the number of design pattern instances that evolve. The third optimization introduces a reward for sequences of minitransformations that were identified by Ó Cinnéide as being capable of creating design pattern instances. This reward is small and constant (i.e., not a percentage based on the individual's QMOOD fitness), and thus provides a small pressure that allows individuals to explore the search space without being unduly penalized. In the fourth experiment, we demonstrated that this reward increases the number, as well as the *type*, of design pattern instances that evolve. The first four experiments helped us to explore and tune the parameters of our experimental setup, and we applied the chosen parameters to a case study in the fifth experiment. The case study applied our approach to a web-based repository for model-driven development artifacts known as REMODD. As the results of the fifth experiment demonstrate, the proposed approach scales to medium-sized designs with tens of classes.

6.2 Impact

We believe that this work has practical applications as a tool for software engineering students and as a design assistance tool for software engineers. Our approach makes suggestions regarding the Gamma design patterns that can be applied to an existing design and provides a well-defined series of refactoring steps to construct those instances. A student who is learning about the concepts design patterns could download a tool based on this approach, and that tool would help to identify portions of the student's software designs that can benefit from the application of design patterns. Similarly, one could implement this approach as a plugin for a software modeling tool (e.g., the Eclipse Modeling Framework). This tool could provide real-time design advice to a software engineer who is considering the use of design patterns to improve maintainability and reusability.

6.3 Future Work

Our ongoing research looks at several open questions related to this work, and we consider these questions in this section.

Our approach is designed to be modular and supports the replacement of its various components. However, identifying the optimal combination of design patterns, metrics, and design change mechanisms for a given problem is nontrivial. Therefore, we are considering other combinations of design patterns, software engineering metrics, and design change mechanisms in order to find the combinations that are most effective for specific problem domains or design needs.

The design change mechanisms used in this work focus exclusively on the construction of design pattern instances. We would like to explore the use of traditional refactorings (e.g., moving a method from one class to another) as design change mechanisms in addition to minitransformations of Ó Cinnéide. It is crucial that

these refactorings are behavior-preserving. This addition might enable the approach to make fine-grained changes to the target software design and, in turn, produce a design of higher quality than what is currently possible.

Similarly, we selected the QMOOD metrics suite as the sole measure of quality for evolving software designs. Since these metrics are the measurement tool that enables the GP to select the best individuals to survive, it follows that our choice of metrics influences the type and number of design pattern instances that evolve. Thus, we are exploring other suites of metrics in order to perform a comparison of *fitness for purpose* from which we can recommend a metrics suite based on the problem domain or software design under consideration.

Fourth, we plan to consider different sets of software engineering metrics, such as those available in DesignAdvisor [3]. Studying a multitude of metrics in this approach will improve our understanding of how sets of refactorings interact with various metrics and design patterns, thus enhancing our ability to tailor the approach to the specific needs of a given problem domain or user.

The experiments presented in this thesis considered only the individual with the greatest fitness in each trial. Since the proposed approach focuses on the introduction of design pattern instances, we plan to modify the GP to examine multiple highly-fit individuals (e.g., the individuals with the five highest unique fitness values in the trial). This will enable us to identify useful individuals that evolved design pattern instances but did not have the highest fitness. During our experimentation for this thesis, we observed useful design pattern instances evolving in individuals that did not have the best fitness, and therefore those instances did not appear in the final results. By exploring a larger group of highly-fit individuals, we may discover richer sets of useful refactorings than what the current implementation provides.

APPENDICES

Appendix A

Queries for Design Pattern Detection

This section contains the Prolog and SQL queries that were developed in order to facilitate detection of design pattern instances. Each query detects a different type of design pattern instance. The queries, along with the type of pattern instance that they detect, are presented in Table A.1.

A.1 PROLOG Queries

Abstract Factory

```
abstract_factory(AFact,CFact,AProd,CProd,Client) :-
      cls(AFact), cls(CFact), cls(AProd), cls(CProd), cls(Client),
      inherit(CFact,AFact), inherit(CProd,AProd),
      instantiate(CFact,CProd), fcall(Client,CProd), AFact \= CFact,
      AProd \= CProd, AFact \= Client, CFact \= Client,
      AProd \= Client, CProd \= Client, AFact \= AProd,
      AProd \= CFact.
```

Adapter

```
adapter(Client, Adapter, Adaptee, Target) :-
    cls(Client), cls(Adapter), cls(Adaptee), cls(Target),
    fcall(Client, Target), inherit(Adapter,Target),
    fcall(Adapter, Adaptee), Client \= Adapter,
    Client \= Adaptee, Client \= Target, Adapter \= Adaptee,
    Adapter \= Target, Adaptee \= Target.
```

Bridge

```
bridge(Abstraction, RefAbstraction, Implementor, ConcImplementor) :-
    inherit(RefAbstraction, Abstraction),
    inherit(ConcImplementor, Implementor),
    aggregate(Abstraction, Implementor),
    Abstraction \= RefAbstraction,
    Abstraction \= Implementor, Abstraction \= ConcImplementor,
    RefAbstraction \= Implementor, RefAbstraction \= ConcImplementor,
    Implementor \= ConcImplementor.
```

Composite

```
composite(Client, Component, Leaf, Composite) :-
    fcall(Client,Component), inherit(Leaf, Component),
    inherit(Composite, Component), aggregate(Composite, Component),
    Client \= Component, Client \= Leaf,
    Client \= Composite, Component \= Leaf,
    Component \= Composite, Leaf \= Composite.
```

Decorator

```
decorator(Component, ConcComponent, Decorator, ConcDecorator) :-
      inherit(ConcComponent, Component), inherit(Decorator, Component),
      inherit(ConcDecorator, Decorator), Component \= ConcComponent,
      Component \= Decorator, Component \= ConcDecorator,
      ConcComponent \= Decorator, ConcComponent \= ConcDecorator,
      Decorator \= ConcDecorator.
```

Prototype

```
prototype(ProtAbs, ProtCon, InstClass, Oper) :-
      cls(ProtAbs), cls(ProtCon), cls(InstClass),
      inherit(ProtCon, ProtAbs), operation(Oper),
      own(ProtAbs,Oper), instantiate(InstClass, ProtAbs),
      ProtAbs \= ProtCon, ProtAbs \= InstClass,
      ProtCon \= InstClass.
```

Proxy

```
proxy(RealSubject, Subject, Proxy) :-
      inherit(RealSubject, Subject), fcall(Proxy, RealSubject),
      RealSubject \= Subject, RealSubject \= Proxy,
      Subject \= Proxy.
```

A.2 SQL Queries

Pattern Type	**SQL Query**
Abstract Factory	SELECT DISTINCT * FROM tClass cAbsProd, tClass cAbsFact, tClass cConProd, tClass cConFact JOIN tInherit tInheritFact ON tInheritFact.sink=cAbsFact.name JOIN tInstantiate ON tInstantiate.source=tInheritFact.source JOIN tInherit tInheritProd ON tInheritProd.source=tInstantiate.sink WHERE tInheritFact.sink != tInheritProd.sink AND tInheritFact.source != tInheritProd.source AND tInheritFact.source = cConFact.name AND tInstantiate.sink = cConProd.name AND tInheritProd.sink = cAbsProd.name
Adapter	SELECT DISTINCT * FROM tClass cClient, tClass cAdapter, tClass cAdaptee, tClass cTarget JOIN tCall callClientTarget ON (cClient.name=callClientTarget.source AND cTarget.name=callClientTarget.sink) JOIN tInherit inheritAdapterTarget ON (inheritAdapterTarget.sink=cTarget.name AND inheritAdapterTarget.source=cAdapter.name) JOIN tCall callAdapterAdaptee ON (callAdapterAdaptee.source=cAdapter.name AND callAdapterAdaptee.sink=cAdaptee.name) WHERE cAdapter.name != cClient.name AND cAdaptee.name != cClient.name AND cAdaptee.name != cAdapter.name AND cTarget.name != cClient.name AND cTarget.name != cAdapter.name AND cTarget.name != cAdaptee.name

Bridge	SELECT DISTINCT * FROM tClass cAbstraction, tClass cRefinedAbstraction, tClass cImplementor, tClass cConcreteImplementor JOIN tInherit inheritRefinedAbstractionAbstraction ON (inheritRefinedAbstractionAbstraction.source=cRefinedAbstraction.name AND inheritRefinedAbstractionAbstraction.sink=cAbstraction.name) JOIN tInherit inheritConcreteImplementor ON (inheritConcreteImplementor.source=cConcreteImplementor.name AND inheritConcreteImplementor.sink=cImplementor.name) JOIN tAggregate aggregateAbstractionImplementor ON (aggregateAbstractionImplementor.source=cAbstraction.name AND aggregateAbstractionImplementor.sink=cImplementor.name) WHERE cRefinedAbstraction.name != cAbstraction.name AND cImplementor.name != cRefinedAbstraction.name AND cConcreteImplementor.name != cAbstraction.name

Composite	SELECT DISTINCT * FROM tClass cClient, tClass cComponent, tClass cLeaf, tClass cComposite JOIN tCall callClientComponent ON (callClientComponent.source=cClient.name AND callClientComponent.sink=cComponent.name) JOIN tInherit inheritLeafComponent ON (inheritLeafComponent.source=cLeaf.name AND inheritLeafComponent.sink=cComponent.name) JOIN tInherit inheritCompositeComponent ON (inheritCompositeComponent.source=cComposite.name AND inheritCompositeComponent.sink=cComponent.name) JOIN tAggregate aggregateCompositeComponent ON (aggregateCompositeComponent.source=cComposite.name AND aggregateCompositeComponent.sink=cComponent.name) WHERE cComponent.name != cClient.name AND cLeaf.name != cClient.name AND cLeaf.name != cComponent.name AND cComposite.name != cClient.name AND cComposite.name != cLeaf.name AND cComposite.name != cComponent.name

Decorator	SELECT DISTINCT * FROM tClass cComponent, tClass cConcreteComponent, tClass cDecorator, tClass cConcreteDecorator JOIN tInherit inheritConcreteComponentComponent ON (inheritConcreteComponentComponent.source=cConcreteComponent.name AND inheritConcreteComponentComponent.sink=cComponent.name) JOIN tInherit inheritDecoratorComponent ON (inheritDecoratorComponent.source=cDecorator.name AND inheritDecoratorComponent.sink=cComponent.name) JOIN tInherit inheritConcreteDecoratorDecorator ON (inheritConcreteDecoratorDecorator.source=cConcreteDecorator.name AND inheritConcreteDecoratorDecorator.sink=cDecorator.name) WHERE cConcreteComponent.name != cComponent.name AND cDecorator.name != cComponent.name AND cDecorator.name != cConcreteComponent.name AND cConcreteDecorator.name != cComponent.name AND cConcreteDecorator.name != cConcreteComponent.name AND cConcreteDecorator.name != cDecorator.name

Prototype	SELECT DISTINCT * FROM tClass protAbs, tClass protCon, tClass cInstClass JOIN tInherit ON tInherit.source=protCon.name JOIN tOwn ON tOwn.source=tInherit.sink JOIN tOperation ON tOperation.name=tOwn.sink JOIN tInstantiate ON (tInstantiate.source=cInstClass.name AND tInstantiate.sink=protAbs.name) WHERE tInherit.sink = protAbs.name AND protAbs.name != protCon.name AND protAbs.name != cInstClass.name AND protCon.name != cInstClass.name
Proxy	SELECT DISTINCT * FROM tClass cRealSubject, tClass cSubject, tClass cProxy JOIN tInherit inheritRealSubjectSubject ON (inheritRealSubjectSubject.source=cRealSubject.name AND inheritRealSubjectSubject.sink=cSubject.name) JOIN tCall callProxyRealSubject ON (callProxyRealSubject.source=cProxy.name AND callProxyRealSubject.sink=cRealSubject.name) WHERE cSubject.name != cRealSubject.name AND cProxy.name != cRealSubject.name AND cSubject.name != cRealSubject.name

Table A.1: SQL Queries for Design Pattern Detection

Appendix B

Transcript of the ReMoDD GP Run

This chapter contains a transcript of the output from our prototype implementation from Experiment V (the REMODD case study) presented in Chapter 5. Following the transcript, we discuss an instance of the `Abstract Factory` design pattern that evolved during the run of Experiment V.

```
Architecture: x86_64


| ECJ
| An evolutionary computation system (version 18)
| By Sean Luke
| URL: http://cs.gmu.edu/~eclab/projects/ecj/
| Mail: ecj-help@cs.gmu.edu
|       (better: join ECJ-INTEREST at URL above)
| Date: June 23, 2008
| Current Java: 1.6.0_14 / Java HotSpot(TM) 64-Bit Server VM-14.0-b16
| Required Minimum Java: 1.3
```

```
Threads:  breed/1 eval/1
Seed: 5
Job: 0
Setting up
Processing GP Types
Processing GP Node Constraints
Processing GP Function Sets
Processing GP Tree Constraints
Determining Tree Sizes
[ ... ]
Compiling Distributions
WARNING:
In function set f0 for the GPTreeConstraints tc0, no *nonterminals*
are given with the return type StringNode which is required by other
functions in the function set or by the tree's return type.  This may
or may not be a problem for you.
PARAMETER: gp.tc.0
WARNING:
In function set f0 for the GPTreeConstraints tc0, no terminals are
given with the return type nil which is required by other functions in
the function set or by the tree's return type.  Nearly all
tree-builders in ECJ require the ability to add a terminal of any type
for which there is a nonterminal, and at any time.  Without terminals,
your code may not work.  One common indication that a tree-builder has
failed due to this problem is if you get the MersenneTwister error 'n
must be positive'.
PARAMETER: gp.tc.0
```

```
Initializing Generation 0
Seed used for refactoring: 5
Input file: Models/ReMoDD.xmi
Tree size penalty: 0.0025
DP Reward Coefficient: 0.5
MT Reward Coefficient: 1.0
Prototype individual set up
[ ... ]
--- Graph construction complete ---
Pattern detector initialized
QMOOD: 3.2699559
Patterns: 136
{ Prototype Prototype Prototype Prototype Prototype Prototype
Prototype Prototype Prototype Prototype Prototype Prototype Prototype
Prototype Prototype Prototype Prototype Prototype Prototype Prototype
Prototype Prototype Prototype Prototype Prototype Prototype Prototype
Prototype Prototype Prototype Prototype Prototype Prototype Prototype
Prototype Prototype Prototype Prototype Prototype Prototype Prototype
Prototype Prototype Prototype Adapter Adapter Adapter Adapter Adapter
Adapter Adapter Adapter Adapter Adapter Adapter Adapter Adapter
Adapter Adapter Adapter Adapter Adapter Adapter Adapter Adapter
Adapter Adapter Adapter Adapter Adapter Adapter Adapter Adapter
Adapter Adapter Adapter Adapter Adapter Adapter Adapter Adapter
Adapter Adapter Adapter Adapter Adapter Adapter Adapter Adapter
Adapter Adapter Adapter Adapter Proxy Proxy Proxy Proxy Proxy Proxy
Proxy Proxy Proxy Proxy Proxy Proxy Proxy Proxy Proxy Proxy Proxy
Proxy Proxy Proxy Proxy Proxy Proxy Decorator Decorator Decorator
```

```
Decorator Decorator Decorator Decorator Decorator Decorator Decorator
Decorator Decorator Decorator Decorator Decorator Decorator Decorator
Decorator Decorator Decorator }
(|V|,|E|): (167,373)
|Classes at start| = 69
Subpop 0 best fitness of generation: Fitness: 8.511621
Generation 1
Subpop 0 best fitness of generation: Fitness: 8.608894
Generation 2
Subpop 0 best fitness of generation: Fitness: 8.195335
Generation 3
Subpop 0 best fitness of generation: Fitness: 8.316859
Generation 4
Subpop 0 best fitness of generation: Fitness: 8.810745
Generation 5
Subpop 0 best fitness of generation: Fitness: 8.3174
Generation 6
Subpop 0 best fitness of generation: Fitness: 8.414059
Generation 7
Subpop 0 best fitness of generation: Fitness: 8.4795265
Generation 8
Subpop 0 best fitness of generation: Fitness: 9.538723
Generation 9
Subpop 0 best fitness of generation: Fitness: 9.022783
Generation 10
Subpop 0 best fitness of generation: Fitness: 9.522734
Generation 11
```

Subpop 0 best fitness of generation: Fitness: 9.737719

Generation 12

Subpop 0 best fitness of generation: Fitness: 9.737986

Generation 13

Subpop 0 best fitness of generation: Fitness: 9.737588

Generation 14

Subpop 0 best fitness of generation: Fitness: 9.737986

Generation 15

Subpop 0 best fitness of generation: Fitness: 9.737986

Generation 16

Subpop 0 best fitness of generation: Fitness: 9.783194

Generation 17

Subpop 0 best fitness of generation: Fitness: 9.786055

Generation 18

Subpop 0 best fitness of generation: Fitness: 9.786582

Generation 19

Subpop 0 best fitness of generation: Fitness: 9.846434

Generation 20

Subpop 0 best fitness of generation: Fitness: 9.847186

Generation 21

Subpop 0 best fitness of generation: Fitness: 9.847186

Generation 22

Subpop 0 best fitness of generation: Fitness: 9.847324

Generation 23

Subpop 0 best fitness of generation: Fitness: 9.845243

Generation 24

Subpop 0 best fitness of generation: Fitness: 9.849785

```
Subpop 0 best fitness of run: Fitness: 9.849785
Subpop 0 best fitness of run: Fitness: 9.849785
Subpop 0 tree size of best individual: 19
Subpop 0 MTs in best individual: 9
Subpop 0 graph (|V|,|E|) of best individual: (173,380)
|Classes at end| = 75
New pattern instances in best individual: 24
```

Sample Design Pattern Instance Next, we consider an instance of the ABSTRACT FACTORY design pattern that evolved during the run whose transcript is shown above. A graphical depiction of this instance is shown in Figure B.1. Each rectangle represents a class in the ReMoDD design. For ease of understanding and post-processing of results, the string "<role>" is appended by the GP to each class that plays a role in the design pattern instances, and only those role-playing classes are shown in the figure. The specific roles and the classes that play them are given in Table B.1. The class `ArrayList` belongs to the Java standard library. We can understand the process that created the `PAbstractRandomWord` class by analyzing its name. First, the "RandomWord" suffix was produced by a `StringNode` that retrieved a random noun from the GP's domain-specific vocabulary that was discussed in Section 5.1.6. This noun was returned to the parent node of the `StringNode`. One of the `StringNode`'s ancestor nodes was a `PartialAbstraction` node that prefixed "RandomWord" with "PAbstract" (an abbreviation of the node type), thus resulting in the final name: "PAbstractRandomWord". This analysis is confirmed by the transformation tree depicted in Figure B.2, which is taken from the best individual of the GP run described in this chapter.

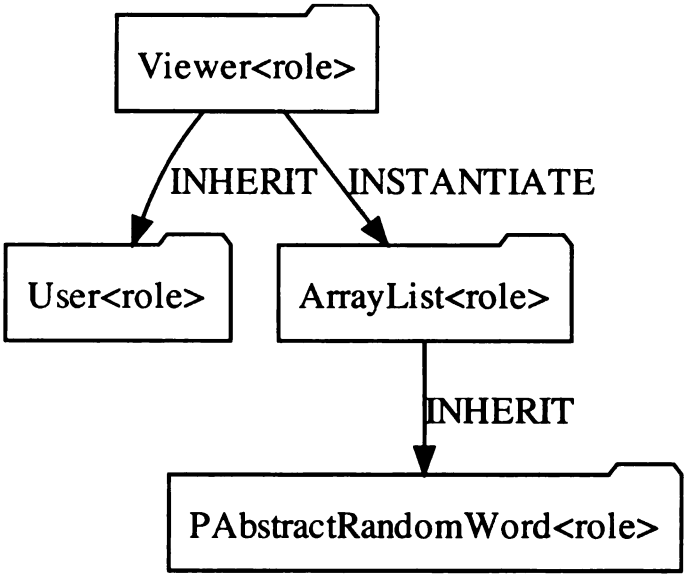


Figure B.1: Sample `Abstract Factory` Design Pattern Instance

Role	**Class Filling the Role**
Abstract Factory	`User`
Concrete Factory	`Viewer`
Abstract Product	`ArrayList`
Concrete Product	`PAbstractRandomWord`

Table B.1: Role-playing Classes in Sample `Abstract Factory` Instance

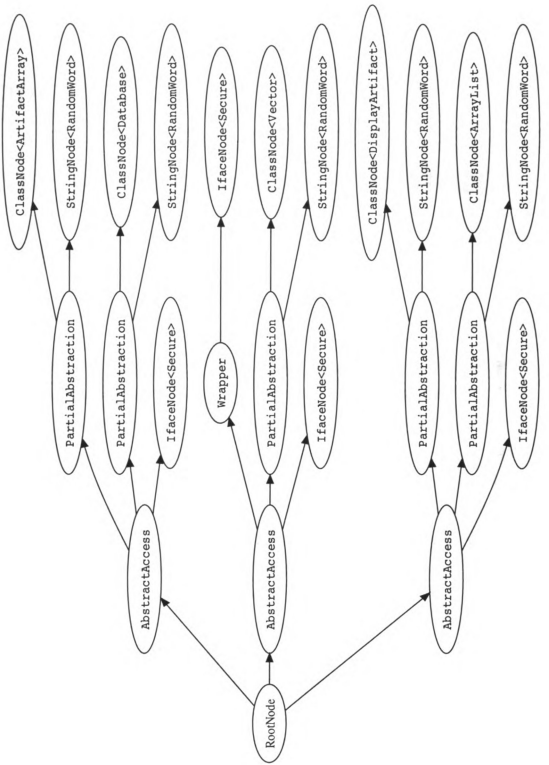


Figure B.2: Transformation Tree for Sample `Abstract Factory` Instance

Appendix C

XMI Representation of ReMoDD Design

<?xml version = '1.0' encoding = 'UTF-8' ?> <XMI xmi.version = '1.2' xmlns:UML =
'org.omg.xmi.namespace.UML' timestamp = 'Fri Jun 12 14:31:38 EDT 2009'>
<XMI.header> <XMI.documentation> <XMI.exporter> </XMI.exporter>
<XMI.exporterVersion> </XMI.exporterVersion> </XMI.documentation> <XMI.metamodel
xmi.name="UML" xmi.version="1.4"/> </XMI.header> <XMI.content> <UML:Model xmi.id
= '-64--88-2-2--3aa132a6:115214597f0:-8000:000000000000077B' name = 'ReMoDD'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Namespace.ownedElement> <UML:Model xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077B' name =
'untitledModel' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:Namespace.ownedElement> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C' name = 'User'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'> <UML:Classifier.feature>
<UML:Attribute xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008A2' name = 'email'
visibility = 'private' isSpecification = 'false' ownerScope = 'instance'

changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008A3'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008A4' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008A5' name = 'name'
visibility = 'private' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008A6'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008A7' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000A9D' name = 'getEmail'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000A9E' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000A9F' name = 'getName'

visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AA0' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AA1' name = 'setEmail'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AA2' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AA3' name = 'email'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AC3' name = 'setName'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AC4' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AC5' name = 'name'

```
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Attribute xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BEC' name =
'institution' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BED'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BEE' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BEF' name = 'purpose'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BF0'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BF1' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BFF' name = 'getPurpose'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
```

'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C00' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C01' name =
'getInstitution' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C02' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C06' name =
'setInstitution' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C07' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C08' name = 'company'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C09' name = 'setPurpose'

visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C0A' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C0B' name = 'purpose'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Attribute xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000BFD' name = 'userName'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000BFE'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000BFF' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C03' name = 'password'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C04'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C05' lower = '1' upper =

'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C06' name = 'getUserName'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C07' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C0A' name = 'getPassword'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C0B' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C0C' name = 'setPassword'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C0D' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =

'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C0E' name = 'pass'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C0F' name = 'setUserName'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C10' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C11' name = 'newname'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E' name = 'Viewer'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000780'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Operation xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AAC' name =
'getAlternateEmails' visibility = 'public' isSpecification = 'false' ownerScope
= 'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false'
isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter>
<UML:Parameter xmi.id =

'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AAD' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AB0'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AB1' name =
'addAlternateEmail' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AB2' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AB3' name = 'email'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AB4' name =
'removeAlternateEmail' visibility = 'public' isSpecification = 'false'
ownerScope = 'instance' isQuery = 'false' concurrency = 'sequential' isRoot =
'false' isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter>
<UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AB5' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AB6' name = 'email'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DB8' name = 'newOperation'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DB9' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Generalization
xmi.id = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000780'
isSpecification = 'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000781' name =
'Contributor' visibility = 'public' isSpecification = 'false' isRoot = 'false'
isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000789'/>
</UML:GeneralizableElement.generalization> </UML:Class> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000785' name = 'Moderator'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078A'/>
</UML:GeneralizableElement.generalization> </UML:Class> <UML:Class xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000787' name =
'SystemAdmin' visibility = 'public' isSpecification = 'false' isRoot = 'false'
isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078D'/>
</UML:GeneralizableElement.generalization> </UML:Class> <UML:Generalization
xmi.id = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000789'
isSpecification = 'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000781'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078A' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000785'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000781'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078D' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000787'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000785'/>
</UML:Generalization.parent> </UML:Generalization> <UML:DataType xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E' name = 'void'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'/> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791' name = 'Artifact'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>

<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =

'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012E5'/>

</UML:GeneralizableElement.generalization> <UML:Classifier.feature>

<UML:Attribute xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B1' name = 'name'

visibility = 'public' isSpecification = 'false' ownerScope = 'instance'

changeability = 'changeable' targetScope = 'instance'>

<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B2'>

<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B3' lower = '1' upper =

'1'/> </UML:Multiplicity.range> </UML:Multiplicity>

</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class

xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>

</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B4' name = 'author'

visibility = 'public' isSpecification = 'false' ownerScope = 'instance'

changeability = 'changeable' targetScope = 'instance'>

<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B5'>

<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B6' lower = '1' upper =

'1'/> </UML:Multiplicity.range> </UML:Multiplicity>

</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class

xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>

</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B7' name =

'description' visibility = 'public' isSpecification = 'false' ownerScope =

'instance' changeability = 'changeable' targetScope = 'instance'>

<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =

```
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B8'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B9' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000848' name = 'type'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000849'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000084A' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000084E' name = 'source'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000084F'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000850' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000851' name = 'quality'
```

visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000852'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000853' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000867' name = 'version'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000868'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000869' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:000000000000C33' name = 'getTitle'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:000000000000C34' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =

'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C35' name = 'getAuthors'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C36' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C37'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C38' name =
'getDescription' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C39' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C3A' name = 'getType'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C3B' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C3C' name = 'getSource'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery

= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C3D' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C3E' name = 'getQuality'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C3F' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C44' name = 'getVersion'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C45' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C46' name = 'setTitle'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C47' name = 'return'

isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C48' name = 'title'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C49' name =
'setDescription' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C4A' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C4E' name = 'desc'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C4B' name = 'setAuthors'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C4C' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =

'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C4D' name = 'authors'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C37'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C4F' name = 'setType'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C50' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C51' name = 'arg2'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C52' name = 'setQuality'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C53' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C5B' name = 'qual'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>

```
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C54' name = 'setSource'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C55' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C56' name = 'src'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Attribute xmi.id =
'-64--88-2-3-67bbd98c:115c40dd9c1:-8000:0000000000000D8B' name = 'owner'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-67bbd98c:115c40dd9c1:-8000:0000000000000D8C'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-67bbd98c:115c40dd9c1:-8000:0000000000000D8D' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CF7' name = 'artifactId'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CF8'>
```

<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CF9' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012BB' name = 'files'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012BC'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012BD' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012BF' name = 'getFiles'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012C0' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012BE'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012C1' name = 'addFile'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =

'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012C2' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012C3' name = 'file'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Attribute xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012D5' name = 'links'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012D6'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012D7' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012D8'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012D9' name =
'securityPolicy' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012DA'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012DB' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D8C'/>

```
</UML:StructuralFeature.type> </UML:Attribute> </UML:Classifier.feature>
</UML:Class> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795' name = 'Review'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'> <UML:Classifier.feature>
<UML:Attribute xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000882' name = 'review'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000883'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000884' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CFA' name = 'artifactId'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CFB'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CFC' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CFD' name = 'version'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
```

```
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CFE'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CFF' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012C4' name = 'approver'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012C5'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012C6' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-18--50--1c3d7122:11620cc66b0:-8000:0000000000000D95' name = 'owner'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--1c3d7122:11620cc66b0:-8000:0000000000000D96'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--1c3d7122:11620cc66b0:-8000:0000000000000D97' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
```

```
</UML:StructuralFeature.type> </UML:Attribute> </UML:Classifier.feature>
</UML:Class> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797' name = 'Petition'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'> <UML:Classifier.feature>
<UML:Attribute xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000895' name = 'description'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000896'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000897' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000898' name = 'state'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000899'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000089A' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:000000000000089B'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CF4' name = 'artifactID'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
```

```
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CF5'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000CF6' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000D00' name = 'userName'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000D01'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000D02' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-18--50--1c3d7122:11620cc66b0:-8000:0000000000000D92' name = 'approver'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--1c3d7122:11620cc66b0:-8000:0000000000000D93'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--1c3d7122:11620cc66b0:-8000:0000000000000D94' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
```

```
</UML:StructuralFeature.type> </UML:Attribute> </UML:Classifier.feature>
</UML:Class> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079B' name =
'EmailSystem' visibility = 'public' isSpecification = 'false' isRoot = 'false'
isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:Classifier.feature> <UML:Operation xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A7' name = 'sendEmail'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A8' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A9' name = 'email'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AD' name =
'sendVerificationEmail' visibility = 'public' isSpecification = 'false'
ownerScope = 'instance' isQuery = 'false' concurrency = 'sequential' isRoot =
'false' isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter>
<UML:Parameter xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AE' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AF' name = 'email'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
```

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D' name = 'Database'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'> <UML:Classifier.feature>
<UML:Attribute xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D8D' name = 'instance'
visibility = 'protected' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D8E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D8F' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008EC' name = 'addArtifact'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008ED' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008EE' name = 'anArtifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>

</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000901' name = 'addReview'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000902' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000903' name = 'aReview'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C22' name =
'addReviewPending' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C23' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C26' name = 'aReview'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000090C' name = 'addSecurity'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery

= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000090D' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000090E' name = 'policy'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000858'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000089C' name =
'addUserPending' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:000000000000089D' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008CD' name = 'aUser'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079F' name = 'addUser'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A0' name = 'return'

isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008CE' name = 'aUser'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB0' name =
'addArtifactLeaf' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB1' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB2' name = 'aLeaf'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D52'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB3' name = 'addPetition'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB4' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =

'35-9

isSpe

'-64-

</UM

</UM

'35-

'del

'ins

= 'f

xmi.

'ret

<UM

'35

</U

'35

isS

'-6

</U

</U

'3

'd

=

is

<U

'3

is

Xm

</

'

is

'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB5' name = 'aPetition'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008F1' name =
'deleteArtifact' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:00000000000008F2' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB6' name = 'anArtifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB7' name =
'deleteArtifactLeaf' visibility = 'public' isSpecification = 'false' ownerScope
= 'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false'
isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter>
<UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB8' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DB9' name = 'aLeaf'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =

'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D52'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008FE' name =
'deletePetition' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:00000000000008FF' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000900' name = 'aPetition'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000909' name = 'deleteReview'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000090A' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000090B' name = 'aReveiw'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =

'35-9-22--98--28a78339:115482af049:-8000:000000000000090F' name =
'deleteSecurity' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:0000000000000910' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000911' name = 'policy'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000858'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A1' name = 'deleteUser'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A2' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008CF' name = 'aUser'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000E9A' name =
'doArtifactSearch' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf

= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000E9B' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012D8'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-531c0b15:1154e2a0a2b:-8000:0000000000000AD5' name = 'theCriteria'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBD'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBA' name =
'doReviewSearch' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBB' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-2-3-f009910:116083ad892:-8000:0000000000000D8B'/> </UML:Parameter.type>
</UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBC' name = 'theCriteria'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBD'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBE' name = 'doUserSearch'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBF' name = 'return'

isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA8'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC0' name = 'theCriteria'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBD'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008EF' name =
'getArtifactByID' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:00000000000008F0' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA5' name = 'artifactId'
isSpecification = 'false'> <UML:Parameter.type> <UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B0'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C0C' name =
'getArtifactLeaf' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C0D' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC1'/>

```
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000C0E' name = 'anArtifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA2' name =
'getOutstandingPetitions' visibility = 'public' isSpecification = 'false'
ownerScope = 'instance' isQuery = 'false' concurrency = 'sequential' isRoot =
'false' isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter>
<UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA3' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA4'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D9F' name =
'getPendingArtifacts' visibility = 'public' isSpecification = 'false' ownerScope
= 'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false'
isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter>
<UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA0' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012D8'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA6' name =
'getPendingUsers' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
```

xmi.id = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA7' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA8'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008FC' name = 'getPetitions'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008FD' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC2'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA1' name = 'anArtifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000001039' name = 'getReview'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:000000000000103A' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-2-3-f009910:116083ad892:-8000:0000000000000D8B'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:000000000000103B' name = 'anArtifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>

</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>

</UML:Operation> <UML:Operation xmi.id =

'35-9-22--98--28a78339:115482af049:-8000:0000000000000912' name = 'getSecurity'

visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery

= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'

isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =

'35-9-22--98--28a78339:115482af049:-8000:0000000000000913' name = 'return'

isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class

xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC3'/>

</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =

'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D9D' name = 'anArtifact'

isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =

'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>

</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>

</UML:Operation> <UML:Operation xmi.id =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A5' name =

'getUserByID' visibility = 'public' isSpecification = 'false' ownerScope =

'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf

= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter

xmi.id = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A6' name =

'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>

<UML:Class xmi.idref =

'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>

</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =

'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC4' name = 'aUserId'

isSpecification = 'false'> <UML:Parameter.type> <UML:DataType xmi.idref =

'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B0'/>

</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>

</UML:Operation> <UML:Operation xmi.id =

'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D90' name = 'instance'

visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D91' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000079D'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-f009910:116083ad892:-8000:0000000000000D84' name = 'queryUsers'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-f009910:116083ad892:-8000:0000000000000D85' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA8'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-f009910:116083ad892:-8000:0000000000000D86' name = 'theCriteria'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBD'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008F3' name =
'updateArtifact' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:00000000000008F4' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =

'35-9-22--98--28a78339:115482af049:-8000:00000000000008F5' name = 'aArtifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008F8' name =
'updatePetition' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:00000000000008F9' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008FB' name = 'aPetition'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC5' name = 'updateReview'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC6' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC7' name = 'aReview'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>

```
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC8' name = 'updateUser'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC9' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DCA' name = 'aUser'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Package xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AA' name = 'java'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Namespace.ownedElement> <UML:Package xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AB' name = 'lang'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Namespace.ownedElement> <UML:Class xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC' name = 'String'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'
isActive = 'false'/> </UML:Namespace.ownedElement> </UML:Package>
</UML:Namespace.ownedElement> </UML:Package> <UML:DataType xmi.id =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B0' name = 'int'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'/> <UML:Association xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000081E' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
```

'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000081F' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000820'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000821' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000822' name =
'Regarding' visibility = 'public' isSpecification = 'false' isNavigable = 'true'
ordering = 'unordered' aggregation = 'none' targetScope = 'instance'
changeability = 'changeable'> <UML:AssociationEnd.multiplicity>
<UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000823'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000824' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000825' name = 'Regarding'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000826' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'

aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000827'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000828' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000829' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000082A'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000082B' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000082C' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000082D' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000082E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =

'-64--

'1'/>

</UML:

xmi.id

</UML

xmi.id

'publ

aggre

<UML:

'-64-

<UML:

'-64-

'1'/>

</UM

xmi.

</UM

</UM

'-64

isSp

'fal

'-64

visi

'unc

'cha

'-64

<UM

'-64

'1'

</U

xmi

'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000082F' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000830' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000831'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000832' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000833' name = 'Submits'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000834' name = 'Reviewer'
visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =
'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000835'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000836' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>

```
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000837' name =
'Review' visibility = 'public' isSpecification = 'false' isNavigable = 'true'
ordering = 'unordered' aggregation = 'none' targetScope = 'instance'
changeability = 'changeable'> <UML:AssociationEnd.multiplicity>
<UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000838'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000839' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000083A' name = 'Submits'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000083B' name = 'petitioner'
visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =
'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000083C'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000083D' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000781'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000083E' name =
'Petition' visibility = 'public' isSpecification = 'false' isNavigable = 'true'
```

ordering = 'unordered' aggregation = 'none' targetScope = 'instance'
changeability = 'changeable'> <UML:AssociationEnd.multiplicity>
<UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000083F'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000840' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000841' name = 'Submits'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000842' name = 'submittor'
visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =
'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000843'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000844' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000781'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000845' name =
'submission' visibility = 'public' isSpecification = 'false' isNavigable =
'true' ordering = 'unordered' aggregation = 'none' targetScope = 'instance'
changeability = 'changeable'> <UML:AssociationEnd.multiplicity>
<UML:Multiplicity xmi.id =

'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000846'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000847' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:DataType xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000857' name = 'boolean'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'/> <UML:Class xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000858' name =
'ArtifactSecurityPolicy' visibility = 'public' isSpecification = 'false' isRoot
= 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012E4'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Attribute xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000884' name = 'emails'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000885'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000886' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000887'/>
</UML:StructuralFeature.type> </UML:Attribute> </UML:Classifier.feature>
</UML:Class> <UML:Association xmi.id =

'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000085A' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000085B' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000881'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000880' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000085E' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000883'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000882' lower = '0' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000858'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000086A' name = 'links to'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000086B' name = '' visibility =

'public' isSpecification = 'false' isNavigable = 'false' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000876'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000875' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000086E' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:000000000000087A'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000879' lower = '0' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000887' name = 'StringArray'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Association xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000087B' name = 'version of'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000087C' visibility = 'public'
isSpecification = 'false' isNavigable = 'false' ordering = 'unordered'

aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008AF'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008AE' lower = '0' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:000000000000087F' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008C9'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008C8' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000888' name = 'Approves'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000889' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000088A'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =

'35-9-22--98--28a78339:115482af049:-8000:000000000000088B' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000785'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:000000000000088C' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000088D'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000088E' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000785'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000089B' name = 'enum'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000089E' name = 'v' visibility
= 'public' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false' isActive = 'false'/> <UML:Association xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008D0' name = 'notifies'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008D1' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =

'35-9-22--98--28a78339:115482af049:-8000:00000000000008D2'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008D3' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:00000000000008D4' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008D5'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008D6' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079B'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008D7' name = 'Approves'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008D8' name = 'Approver'
visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =
'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008E5'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008E4' lower = '1' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>

</U

xm

</U

xm

'O

or

ch

<U

'3

<U

'3

'1

</U

xm

</U

</U

'3

'A

=

<U

'3

'C

is

'3

is

'f

'3

<U

'3

'1

```
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000787'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:00000000000008DB' name =
'0...*' visibility = 'public' isSpecification = 'false' isNavigable = 'true'
ordering = 'unordered' aggregation = 'none' targetScope = 'instance'
changeability = 'changeable'> <UML:AssociationEnd.multiplicity>
<UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008DC'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008DD' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000915' name =
'ArtifactSecurutyPolicy' visibility = 'public' isSpecification = 'false' isRoot
= 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/>
<UML:Collaboration xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000919' name =
'Collaboration' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:Namespace.ownedElement> <UML:ClassifierRole xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091A' name = 'aBrowser'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:ClassifierRole.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091C'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091B' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
```

```
</UML:ClassifierRole.multiplicity> <UML:ClassifierRole.base> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>
</UML:ClassifierRole.base> </UML:ClassifierRole> <UML:ClassifierRole xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091D' name = 'aReview'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:ClassifierRole.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091F'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091E' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:ClassifierRole.multiplicity> <UML:ClassifierRole.base> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:ClassifierRole.base> <UML:ClassifierRole.availableFeature> <UML:Attribute
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:0000000000000882'/>
</UML:ClassifierRole.availableFeature> </UML:ClassifierRole>
<UML:AssociationRole xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000921' isSpecification =
'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'>
<UML:Association.connection> <UML:AssociationEndRole xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000922' isSpecification =
'false' isNavigable = 'false'> <UML:AssociationEnd.participant>
<UML:ClassifierRole xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091A'/>
</UML:AssociationEnd.participant> </UML:AssociationEndRole>
<UML:AssociationEndRole xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000923' isSpecification =
'false' isNavigable = 'false'> <UML:AssociationEnd.participant>
<UML:ClassifierRole xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091D'/>
</UML:AssociationEnd.participant> </UML:AssociationEndRole>
```

</UML:Association.connection> <UML:AssociationRole.message> <UML:Message
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:0000000000000925'/>
</UML:AssociationRole.message> </UML:AssociationRole> <UML:ClassifierRole xmi.id
= '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000973' name = 'theDB'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:ClassifierRole.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000975'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000974' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:ClassifierRole.multiplicity> <UML:ClassifierRole.base> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D'/>
</UML:ClassifierRole.base> <UML:ClassifierRole.availableFeature> <UML:Operation
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079F'/>
<UML:Operation xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A1'/> <UML:Operation
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A5'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000089C'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:00000000000008EC'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008EF'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:00000000000008F1'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008F3'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:00000000000008F8'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008FC'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:00000000000008FE'/>
<UML:Operation xmi.idref =

'35-9-22--98--28a78339:115482af049:-8000:0000000000000901'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:0000000000000909'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000090C'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:000000000000090F'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000912'/>
</UML:ClassifierRole.availableFeature> </UML:ClassifierRole>
<UML:AssociationRole xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000977' isSpecification =
'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'>
<UML:Association.connection> <UML:AssociationEndRole xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000978' isSpecification =
'false' isNavigable = 'false'> <UML:AssociationEnd.participant>
<UML:ClassifierRole xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091D'/>
</UML:AssociationEnd.participant> </UML:AssociationEndRole>
<UML:AssociationEndRole xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000979' isSpecification =
'false' isNavigable = 'false'> <UML:AssociationEnd.participant>
<UML:ClassifierRole xmi.idref =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000973'/>
</UML:AssociationEnd.participant> </UML:AssociationEndRole>
</UML:Association.connection> <UML:AssociationRole.message> <UML:Message
xmi.idref = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000097A'/>
</UML:AssociationRole.message> </UML:AssociationRole> <UML:ClassifierRole xmi.id
= '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000980' name = 'theDB'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:ClassifierRole.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000982'>

<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000981' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:ClassifierRole.multiplicity> <UML:ClassifierRole.base> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D'/>
</UML:ClassifierRole.base> <UML:ClassifierRole.availableFeature> <UML:Operation
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079F'/>
<UML:Operation xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A1'/> <UML:Operation
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007A5'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000089C'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:00000000000008EC'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008EF'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:00000000000008F1'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008F3'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:00000000000008F8'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:00000000000008FC'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:00000000000008FE'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000901'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:0000000000000909'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000090C'/> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:000000000000090F'/>
<UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000912'/>

```
</UML:ClassifierRole.availableFeature> </UML:ClassifierRole>
<UML:AssociationRole xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000984' isSpecification =
'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'>
<UML:Association.connection> <UML:AssociationEndRole xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000985' name = ''
isSpecification = 'false' isNavigable = 'false'>
<UML:AssociationEnd.participant> <UML:ClassifierRole xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091D'/>
</UML:AssociationEnd.participant> </UML:AssociationEndRole>
<UML:AssociationEndRole xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000986' isSpecification =
'false' isNavigable = 'false'> <UML:AssociationEnd.participant>
<UML:ClassifierRole xmi.idref =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000980'/>
</UML:AssociationEnd.participant> </UML:AssociationEndRole>
</UML:Association.connection> <UML:AssociationRole.message> <UML:Message
xmi.idref = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000987'/>
</UML:AssociationRole.message> </UML:AssociationRole>
</UML:Namespace.ownedElement> <UML:Collaboration.interaction> <UML:Interaction
xmi.id = '35-9-22--98--28a78339:115482af049:-8000:0000000000000924' name =
'newInteraction' isSpecification = 'false'> <UML:Interaction.message>
<UML:Message xmi.id = '35-9-22--98--28a78339:115482af049:-8000:0000000000000925'
name = 'writeReview()' isSpecification = 'false'> <UML:Message.sender>
<UML:ClassifierRole xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091A'/>
</UML:Message.sender> <UML:Message.receiver> <UML:ClassifierRole xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091D'/>
</UML:Message.receiver> <UML:Message.communicationConnection>
<UML:AssociationRole xmi.idref =
```

'35-9-22--98--28a78339:115482af049:-8000:0000000000000921'/>
</UML:Message.communicationConnection> <UML:Message.action> <UML:CreateAction
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:0000000000000920'/>
</UML:Message.action> </UML:Message> <UML:Message xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000097A' isSpecification =
'false'> <UML:Message.activator> <UML:Message xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000925'/>
</UML:Message.activator> <UML:Message.sender> <UML:ClassifierRole xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091D'/>
</UML:Message.sender> <UML:Message.receiver> <UML:ClassifierRole xmi.idref =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000973'/>
</UML:Message.receiver> <UML:Message.communicationConnection>
<UML:AssociationRole xmi.idref =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000977'/>
</UML:Message.communicationConnection> <UML:Message.action> <UML:CreateAction
xmi.idref = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000976'/>
</UML:Message.action> </UML:Message> <UML:Message xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000987' name =
'addReview()' isSpecification = 'false'> <UML:Message.activator> <UML:Message
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:0000000000000925'/>
</UML:Message.activator> <UML:Message.sender> <UML:ClassifierRole xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000091D'/>
</UML:Message.sender> <UML:Message.receiver> <UML:ClassifierRole xmi.idref =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000980'/>
</UML:Message.receiver> <UML:Message.predecessor> <UML:Message xmi.idref =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000097A'/>
</UML:Message.predecessor> <UML:Message.communicationConnection>
<UML:AssociationRole xmi.idref =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000984'/>
</UML:Message.communicationConnection> <UML:Message.action> <UML:CallAction

```
xmi.idref = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000983'/>
</UML:Message.action> </UML:Message> </UML:Interaction.message>
</UML:Interaction> </UML:Collaboration.interaction> </UML:Collaboration>
<UML:Class xmi.id = '35-9-22--98--28a78339:115482af049:-8000:0000000000000926'
name = '' visibility = 'public' isSpecification = 'false' isRoot = 'false'
isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:Namespace.ownedElement> <UML:StateMachine xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000927' isSpecification =
'false'> <UML:StateMachine.context> <UML:Class xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000926'/>
</UML:StateMachine.context> <UML:StateMachine.top> <UML:CompositeState xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000928' name = 'top'
isSpecification = 'false' isConcurrent = 'false'> <UML:CompositeState.subvertex>
<UML:SimpleState xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:000000000000092A' isSpecification =
'false'> <UML:State.doActivity> <UML:CreateAction xmi.id =
'35-9-22--98--28a78339:115482af049:-8000:0000000000000920' name = 'new Review'
isSpecification = 'false' isAsynchronous = 'false'>
<UML:CreateAction.instantiation> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:CreateAction.instantiation> </UML:CreateAction> </UML:State.doActivity>
</UML:SimpleState> </UML:CompositeState.subvertex> </UML:CompositeState>
</UML:StateMachine.top> </UML:StateMachine> </UML:Namespace.ownedElement>
</UML:Class> <UML:TagDefinition xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000096A' isSpecification =
'false' tagType = 'String'> <UML:TagDefinition.multiplicity> <UML:Multiplicity
xmi.id = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000096B'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000096C' lower = '0' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
```

```
</UML:TagDefinition.multiplicity> </UML:TagDefinition> <UML:TagDefinition xmi.id
= '-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000096D' isSpecification =
'false' tagType = 'String'> <UML:TagDefinition.multiplicity> <UML:Multiplicity
xmi.id = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000096E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000096F' lower = '0' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:TagDefinition.multiplicity> </UML:TagDefinition> <UML:TagDefinition xmi.id
= '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000970' isSpecification =
'false' tagType = 'String'> <UML:TagDefinition.multiplicity> <UML:Multiplicity
xmi.id = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000971'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000972' lower = '0' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:TagDefinition.multiplicity> </UML:TagDefinition> <UML:Class xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000097B' name = ''
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'> <UML:Namespace.ownedElement>
<UML:StateMachine xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000097C' isSpecification =
'false'> <UML:StateMachine.context> <UML:Class xmi.idref =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000097B'/>
</UML:StateMachine.context> <UML:StateMachine.top> <UML:CompositeState xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000097D' name = 'top'
isSpecification = 'false' isConcurrent = 'false'> <UML:CompositeState.subvertex>
<UML:SimpleState xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000097F' isSpecification =
'false'> <UML:State.doActivity> <UML:CreateAction xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000976' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
```

<UML:SimpleState xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000988' isSpecification =
'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000983' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
</UML:CompositeState.subvertex> </UML:CompositeState> </UML:StateMachine.top>
</UML:StateMachine> </UML:Namespace.ownedElement> </UML:Class> <UML:Association
xmi.id = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000990' name =
'Created By' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000991' name = 'creation'
visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =
'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000F1E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000F1D' lower = '0' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000994' visibility
= 'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000995'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000996' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class



```
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000997' name = 'Stores'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:0000000000000998' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000F3E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000F3D' lower = '0' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000099B' visibility
= 'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000EFA'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000EF9' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000099E' name = 'stores'
```

isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:000000000000099F' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000F40'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000F3F' lower = '0' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:00000000000009A2' visibility
= 'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000EFE'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000EFD' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:00000000000009A5' name = 'stores'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:00000000000009A6' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'

aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000F64'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000F63' lower = '0' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-18--49--d903a0f:11548fd9b5e:-8000:00000000000009A9' visibility
= 'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:00000000000009AA'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:00000000000009AB' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009AD' name = 'Login'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CD'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC9'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Operation xmi.id =
'35-9-22--98-531c0b15:1154e2a0a2b:-8000:0000000000000ADA' name = 'logout'

visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id = '35-9-22--98-531c0b15:1154e2a0a2b:-8000:0000000000000ADB' name = 'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/> </UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter> </UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Class xmi.id = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009AF' name = 'Search' visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'> <UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref = '35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CE'/> <UML:Generalization xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCB'/> </UML:GeneralizableElement.generalization> <UML:Classifier.feature> <UML:Operation xmi.id = '-64--88-2-2--621e829:115791c7bbc:-8000:0000000000000CB3' name = 'doArtifactSearch' visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id = '-64--88-2-2--621e829:115791c7bbc:-8000:0000000000000CB4' name = 'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class xmi.idref = '-64--88-2-2-1994c0fa:1157b18466d:-8000:0000000000000BE8'/> </UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id = '-64--88-2-2--621e829:115791c7bbc:-8000:0000000000000CB5' name = 'searchString' isSpecification = 'false'> <UML:Parameter.defaultValue> <UML:Expression xmi.id = '-64--88-2-3-4b35a1de:116080df201:-8000:00000000000012BB' language = 'UML 1.4' body = ''''/> </UML:Parameter.defaultValue> <UML:Parameter.type> <UML:Class xmi.idref =

```
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DAA' name = 'type'
isSpecification = 'false'> <UML:Parameter.defaultValue> <UML:Expression xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000012BC' language = 'UML 1.4'
body = '&apos;all&apos;'/> </UML:Parameter.defaultValue> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DAC' name = 'source'
isSpecification = 'false' kind = 'in'> <UML:Parameter.defaultValue>
<UML:Expression xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000012BD' language = 'UML 1.4'
body = '&apos;all&apos;'/> </UML:Parameter.defaultValue> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DAE' name = 'quality'
isSpecification = 'false' kind = 'in'> <UML:Parameter.defaultValue>
<UML:Expression xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000012BE' language = 'UML 1.4'
body = '&apos;all&apos;'/> </UML:Parameter.defaultValue> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Class xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B1' name = 'ManageUsers'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
```

'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CF'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD4'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Operation xmi.id =
'-64--88-2-2--621e829:115791c7bbc:-8000:0000000000000C51' name =
'assessApplicant' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-2-2--621e829:115791c7bbc:-8000:0000000000000C52' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C91' name = 'approveUser'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C92' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C96' name = 'aUser'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C93' name = 'rejectUser'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'

isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C94' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C95' name = 'aUser'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Class xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B3' name =
'DisplayArtifact' visibility = 'public' isSpecification = 'false' isRoot =
'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009D0'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCC'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Operation xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BE5' name =
'downloadArtifact' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BE6' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C19' name = 'writeReview'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery

= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C1A' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C1D' name = 'str'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Class xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B7' name = 'EditArtifact'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B9'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD3'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Operation xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BD' name =
'petitionForReconsideration' visibility = 'public' isSpecification = 'false'
ownerScope = 'instance' isQuery = 'false' concurrency = 'sequential' isRoot =
'false' isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter>
<UML:Parameter xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BE' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000BF8' name =

'submitArtifact' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000BF9' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000BFA' name = 'artifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Generalization
xmi.id = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B9'
isSpecification = 'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B7'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B3'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Class xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BA' name =
'AssessArtifacts' visibility = 'public' isSpecification = 'false' isRoot =
'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:ModelElement.clientDependency> <UML:Abstraction xmi.idref =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD0'/>
</UML:ModelElement.clientDependency> <UML:GeneralizableElement.generalization>
<UML:Generalization xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BC'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD2'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Operation xmi.id =

'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BF9' name =
'approveArtifact' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BFA' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BFB' name = 'anArtifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BFC' name =
'rejectArtifact' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BFD' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BFE' name = 'anArtifact'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E55' name =
'approvePetition' visibility = 'public' isSpecification = 'false' ownerScope =

'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E56' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E57' name = 'aPetition'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E58' name =
'rejectPetition' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E59' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E5A' name = 'aPetition'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Generalization
xmi.id = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BC'
isSpecification = 'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BA'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =

```
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B7'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Association xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BF' name = 'Creates'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009C0' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009C1'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009C2' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B7'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009C3' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000001010'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:000000000000100F' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB' name = 'WebPage'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
```

'false' isAbstract = 'false' isActive = 'false'> <UML:Classifier.feature>
<UML:Operation xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BE7' name = 'display'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-18--49-5dd0b03f:1157c9445a9:-8000:0000000000000BE8' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Generalization
xmi.id = '35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CD'
isSpecification = 'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009AD'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CE' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009AF'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CF' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B1'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009D0' isSpecification =

```
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B3'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Class xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009D2' name = 'Session'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'> <UML:Classifier.feature>
<UML:Operation xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AC6' name = 'setUser'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AC7' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AC8' name = 'user'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Attribute xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AC9' name = 'user'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000ACA'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000ACB' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
```

<
x
<
'
v
=
i
'
i
x
<
<
'
'
'
<
'
<
'
'
'
<
x
<
<
'
j
'
'
=
a

</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/> </UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id = '35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000ACC' name = 'getUser' visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id = '35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000ACD' name = 'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/> </UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter> </UML:Operation> <UML:Attribute xmi.id = '35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000D03' name = 'idleTimeAllowed' visibility = 'public' isSpecification = 'false' ownerScope = 'instance' changeability = 'changeable' targetScope = 'instance'> <UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id = '35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000D04'> <UML:Multiplicity.range> <UML:MultiplicityRange xmi.id = '35-9-22--98--7ff8a111:115e8aac4e9:-8000:0000000000000D05' lower = '1' upper = '1'/> </UML:Multiplicity.range> </UML:Multiplicity> </UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:DataType xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B0'/> </UML:StructuralFeature.type> </UML:Attribute> </UML:Classifier.feature> </UML:Class> <UML:Association xmi.id = '35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A09' name = 'Queries' isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id = '35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A0A' name = '' visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered' aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>

<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000EF8'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000EF7' lower = '1' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A0D' name =
'Creator' visibility = 'public' isSpecification = 'false' isNavigable = 'true'
ordering = 'unordered' aggregation = 'none' targetScope = 'instance'
changeability = 'changeable'> <UML:AssociationEnd.multiplicity>
<UML:Multiplicity xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A0E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A0F' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079D'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A11' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A12' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A13'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =

'35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A14' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A15' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A16'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:0000000000000A17' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009D2'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000000E9C' name = 'ArtifactArray'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:000000000000103C' name = 'Array'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:000000000000103D' name = 'ReviewArray'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AB0' name = 'ArrayList'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Association xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000ACE' name = ''

isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000ACF' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AD0'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AD1' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009D2'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AD2' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AD3'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AD4' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'-64--88-2-2-1994c0fa:1157b18466d:-8000:0000000000000BE8' name =
'List<Artifact>' visibility = 'public' isSpecification = 'false' isRoot =
'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/> <UML:Class
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C12' name =
'AssessReviews' visibility = 'public' isSpecification = 'false' isRoot = 'false'

is

<U

'3

xm

</

<U

'3

'g

'i

=

xm

'r

<U

'-

</

<U

'-

vi

=

is

'-

is

xm

</

'-

is

'-

</

</

'-

```
isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C1E'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCE'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Operation xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C1F' name =
'gerPendingReviews' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C20' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-2-3-f009910:116083ad892:-8000:0000000000000D8B'/> </UML:Parameter.type>
</UML:Parameter> </UML:BehavioralFeature.parameter> </UML:Operation>
<UML:Operation xmi.id =
'-64--88-2-3-23172210:1161df0d9a2:-8000:0000000000000D8B' name = 'approveReview'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-23172210:1161df0d9a2:-8000:0000000000000D8C' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-23172210:1161df0d9a2:-8000:0000000000000D8D' name = 'aReview'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-23172210:1161df0d9a2:-8000:0000000000000D8E' name = 'rejectReview'
```

visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id = '-64--88-2-3-23172210:1161df0d9a2:-8000:0000000000000D8F' name = 'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/> </UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id = '-64--88-2-3-23172210:1161df0d9a2:-8000:0000000000000D90' name = 'aReview' isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/> </UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter> </UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Class xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C14' name = 'ReviewArtifactPage' visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/> <UML:Generalization xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C1E' isSpecification = 'false'> <UML:Generalization.child> <UML:Class xmi.idref = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C12'/> </UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref = '35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/> </UML:Generalization.parent> </UML:Generalization> <UML:Class xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C21' name = 'Reviews' visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/> <UML:Association xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C28' name = 'Evaluates' isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C29' name = 'Evaluator' visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =

'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C30'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C2F' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C12'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C2C' name =
'Evaluatee' visibility = 'public' isSpecification = 'false' isNavigable = 'true'
ordering = 'unordered' aggregation = 'none' targetScope = 'instance'
changeability = 'changeable'> <UML:AssociationEnd.multiplicity>
<UML:Multiplicity xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C32'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C31' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C37' name =
'ArrayList<String>' visibility = 'public' isSpecification = 'false' isRoot
= 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/>
<UML:Association xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C61' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C62' visibility = 'public'

```
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C63'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C64' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C65' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C66'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C67' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C68' name = 'links to'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C69' visibility = 'public'
isSpecification = 'false' isNavigable = 'false' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C70'>
```

<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C6F' lower = '0' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C6C' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000E1B'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000E1A' lower = '0' upper =
'-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C5D' name = 'displays'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C5E' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C5F'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C60' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class

```
xmi.idref = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B3'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C61' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C62'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C63' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C64' name = 'Assesses'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C65' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C66'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C67' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B1'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C68' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
```

```
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C69'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-1b63d4c6:115ac040095:-8000:0000000000000C6A' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-18--49-4c3f2cbc:115b550b35a:-8000:0000000000000D19' name = 'notifies'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-18--49-4c3f2cbc:115b550b35a:-8000:0000000000000D1A' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--49-4c3f2cbc:115b550b35a:-8000:0000000000000D1B'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--49-4c3f2cbc:115b550b35a:-8000:0000000000000D1C' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B1'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-18--49-4c3f2cbc:115b550b35a:-8000:0000000000000D1D' visibility
= 'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--49-4c3f2cbc:115b550b35a:-8000:0000000000000D1E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
```

'-64--88-18--49-4c3f2cbc:115b550b35a:-8000:0000000000000D1F' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079B'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267' name =
'ArtifactComponent' visibility = 'public' isSpecification = 'false' isRoot =
'false' isLeaf = 'false' isAbstract = 'true' isActive = 'false'>
<UML:Classifier.feature> <UML:Attribute xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:000000000000129F' name =
'parentArtifactId' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A0'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A1' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A2' name = 'version'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A3'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A4' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class

```
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A5' name =
'description' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A6'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A7' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A8' name =
'setVersion' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012A9' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012AA' name = 'ver'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012B4' name =
'setDescription' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
```

= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012B5' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012B6' name = 'desc'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E63' name =
'getVersion' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E64' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D74' name = 'add'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D75' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D76' name = 'file'

isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D77' name = 'remove'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D78' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D79' name = 'file'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D7A' name = 'getChild'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D7B' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D7C' name = 'childId'
isSpecification = 'false'> <UML:Parameter.type> <UML:DataType xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007B0'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =

'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D8A' name =
'getDescription' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D8B' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Association xmi.id
= '-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001270' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001271' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'composite' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001278'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001277' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001274'
visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =
'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:000000000000127A'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =

'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001279' lower = '1' upper
= '-1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:000000000000127B' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:000000000000127C' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'aggregate' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:000000000000127D'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:000000000000127E' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-18--50--7c832930:115fc7293e1:-8000:000000000000127F'
visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =
'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001280'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001281' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>

```
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012BE' name =
'ArrayList&lt;ArtifactFiles&gt;' visibility = 'public' isSpecification = 'false'
isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/>
<UML:Class xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012D8' name =
'ArrayList&lt;Artifact&gt;' visibility = 'public' isSpecification = 'false'
isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/>
<UML:Class xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012DC' name =
'ArrayList&lt;ArtifcactSecurityPolicy&gt;' visibility = 'public' isSpecification
= 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive =
'false'/> <UML:Interface xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012DF' name = 'Secure'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false'> <UML:Classifier.feature> <UML:Operation xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012E1' name =
'allowAccess' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' isQuery = 'false' concurrency = 'sequential' isRoot = 'false' isLeaf
= 'false' isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter
xmi.id = '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012E2' name =
'return' isSpecification = 'false' kind = 'return'> <UML:Parameter.type>
<UML:DataType xmi.idref =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000857'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012E3' name = 'aUser'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
```

```
</UML:Operation> </UML:Classifier.feature> </UML:Interface> <UML:Generalization
xmi.id = '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012E4'
isSpecification = 'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000858'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Interface xmi.idref
= '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012DF'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012E5' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Interface xmi.idref
= '-64--88-18--50--7c832930:115fc7293e1:-8000:00000000000012DF'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Class xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E5B' name = 'Petiition'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Association xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E5C' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E5D' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E5E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E5F' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000079B'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
```

xmi.id = '-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E60'
visibility = 'public' isSpecification = 'false' isNavigable = 'true' ordering =
'unordered' aggregation = 'none' targetScope = 'instance' changeability =
'changeable'> <UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E61'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-18--50--3df78d7b:115fc9a075f:-8000:0000000000001E62' lower = '1' upper
= '1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BA'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D52' name = 'ArtifactLeaf'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D81'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Attribute xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D56' name =
'serverFilename' visibility = 'public' isSpecification = 'false' ownerScope =
'instance' changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D57'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D58' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =

```
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D59' name = 'filename'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D5A'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D5B' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D5C' name = 'fileformat'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D5D'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D5E' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Attribute xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D5F' name = 'approver'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D60'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D61' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
```

```
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:StructuralFeature.type> </UML:Attribute> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D62' name = 'getFileFormat'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D63' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D64' name = 'getFilename'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D65' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D66' name = 'getApprover'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D67' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
```

'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D68' name = 'setFileFormat'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D69' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D6A' name = 'fileform'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D6B' name = 'setFilename'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D6C' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D6D' name = 'filename'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> <UML:Operation xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D6E' name = 'setApprover'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance' isQuery
= 'false' concurrency = 'sequential' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:BehavioralFeature.parameter> <UML:Parameter xmi.id =

```
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D6F' name = 'return'
isSpecification = 'false' kind = 'return'> <UML:Parameter.type> <UML:DataType
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000078E'/>
</UML:Parameter.type> </UML:Parameter> <UML:Parameter xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D70' name = 'name'
isSpecification = 'false'> <UML:Parameter.type> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:00000000000007AC'/>
</UML:Parameter.type> </UML:Parameter> </UML:BehavioralFeature.parameter>
</UML:Operation> </UML:Classifier.feature> </UML:Class> <UML:Class xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D54' name =
'ArtifactComposite' visibility = 'public' isSpecification = 'false' isRoot =
'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D82'/>
</UML:GeneralizableElement.generalization> <UML:Classifier.feature>
<UML:Attribute xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D7D' name = 'children'
visibility = 'public' isSpecification = 'false' ownerScope = 'instance'
changeability = 'changeable' targetScope = 'instance'>
<UML:StructuralFeature.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D7E'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D7F' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:StructuralFeature.multiplicity> <UML:StructuralFeature.type> <UML:Class
xmi.idref = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D80'/>
</UML:StructuralFeature.type> </UML:Attribute> </UML:Classifier.feature>
</UML:Class> <UML:Class xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D80' name =
'ArrayList&lt;ArtifactComponent&gt;' visibility = 'public' isSpecification =
```

'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive =
'false'/> <UML:Generalization xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D81' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D52'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D82' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D54'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Association xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D83' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D84' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'composite' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D85'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D86' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D54'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D87' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'

aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D88'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D89' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D8C' name =
'ArrayList<Secure>' visibility = 'public' isSpecification = 'false' isRoot
= 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/> <UML:Class
xmi.id = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D9B' name =
'ArtifactId' visibility = 'public' isSpecification = 'false' isRoot = 'false'
isLeaf = 'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000D9E' name =
'SecurityPolicy' visibility = 'public' isSpecification = 'false' isRoot =
'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/> <UML:Class
xmi.id = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA4' name =
'ArrayList<Petitions>' visibility = 'public' isSpecification = 'false'
isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/>
<UML:Class xmi.id = '-64--88-2-3-4b35a1de:116080df201:-8000:0000000000000DA8'
name = 'ArrayList<User>' visibility = 'public' isSpecification = 'false'
isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/>
<UML:Association xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000011C4' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000011C5' visibility = 'public'

isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000011C6'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000011C7' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '-64--88-2-3-4b35a1de:116080df201:-8000:00000000000011C8' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000011C9'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'-64--88-2-3-4b35a1de:116080df201:-8000:00000000000011CA' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--50--7c832930:115fc7293e1:-8000:0000000000001267'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Class xmi.id =
'-64--88-2-3-f009910:116083ad892:-8000:0000000000000D8B' name =
'ArrayList<Review>' visibility = 'public' isSpecification = 'false' isRoot
= 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/> <UML:Class
xmi.id = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DAF' name = 'bool'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DBD' name = 'Criteria'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =

'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC1' name =
'ArrayList<ArtifactLeaf>' visibility = 'public' isSpecification = 'false'
isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/>
<UML:Class xmi.id = '35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC2'
name = 'ArrayList<Petition>' visibility = 'public' isSpecification =
'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false' isActive =
'false'/> <UML:Class xmi.id =
'35-9-22--98-25f82166:1162298cc26:-8000:0000000000000DC3' name =
'ArtifactSecurity' visibility = 'public' isSpecification = 'false' isRoot =
'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/> <UML:Class
xmi.id = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DAC' name =
'ManageSecurityPolicies' visibility = 'public' isSpecification = 'false' isRoot
= 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD5'/>
</UML:GeneralizableElement.generalization> </UML:Class> <UML:Class xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DAE' name = 'SubmitArtifact'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DBD'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCD'/>
</UML:GeneralizableElement.generalization> </UML:Class> <UML:Class xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB0' name = 'AssessPetitions'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCF'/>
</UML:GeneralizableElement.generalization> </UML:Class> <UML:Class xmi.id =

'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB2' name = 'AssessReviews'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB7' name = 'Register'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DBC'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCA'/>
</UML:GeneralizableElement.generalization> </UML:Class> <UML:Class xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB9' name = 'Homepage'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'>
<UML:GeneralizableElement.generalization> <UML:Generalization xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DBB'/> <UML:Generalization
xmi.idref = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC8'/>
</UML:GeneralizableElement.generalization> </UML:Class> <UML:Generalization
xmi.id = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DBB' isSpecification
= 'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB9'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DBC' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB7'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DBD' isSpecification =

'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DAE'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Association xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DBE' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DBF' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC0'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC1' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DAE'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC2' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC3'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC4' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>

```
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC5' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC6' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC7'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC8' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DAE'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DC9' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DCA'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DCB' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DBA' name = 'creates'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
```

'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DBB' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DBC'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DBD' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB7'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DBE' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DBF'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC0' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077C'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC1' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC2' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =

'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC3'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC4' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000785'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC5' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC6'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC7' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000795'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC8' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB9'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DC9' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009AD'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>

```
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCA' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB7'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCB' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009AF'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCC' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B3'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCD' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DAE'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCE' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'35-9-22--98--4283c415:1157cc6a09d:-8000:0000000000000C12'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
```

```
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DCF' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB0'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Stereotype xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD1' name = 'realize'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Stereotype.baseClass> </UML:Stereotype.baseClass>
</UML:Stereotype> <UML:Abstraction xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD0' isSpecification =
'false'> <UML:ModelElement.stereotype> <UML:Stereotype xmi.idref =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD1'/>
</UML:ModelElement.stereotype> <UML:Dependency.client> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BA'/>
</UML:Dependency.client> <UML:Dependency.supplier> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Dependency.supplier> </UML:Abstraction> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD2' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BA'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD3' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B7'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
```

```
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD4' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B1'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Generalization xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD5' isSpecification =
'false'> <UML:Generalization.child> <UML:Class xmi.idref =
'127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DAC'/>
</UML:Generalization.child> <UML:Generalization.parent> <UML:Class xmi.idref =
'35-9-22--98--11c7cf65:1154d63a10d:-8000:00000000000009CB'/>
</UML:Generalization.parent> </UML:Generalization> <UML:Association xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD6' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD7' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD8'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DD9' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DB0'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DDA' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
```

<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DDB'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DDC' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000797'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DDD' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DDE' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DDF'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE0' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009BA'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE1' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE2'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE3' lower = '1' upper =

```
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE4' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE5' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE6'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE7' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-2-56cf0fde:1154c8d17ce:-8000:00000000000009B7'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE8' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DE9'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DEA' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:0000000000000791'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
```

```
</UML:Association.connection> </UML:Association> <UML:Association xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DEB' name = ''
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Association.connection> <UML:AssociationEnd xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DEC' visibility = 'public'
isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DED'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DEE' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '127-0-1-1-3c94a2b7:116a6b4af4c:-8000:0000000000000DAC'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd> <UML:AssociationEnd
xmi.id = '127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DEF' visibility =
'public' isSpecification = 'false' isNavigable = 'true' ordering = 'unordered'
aggregation = 'none' targetScope = 'instance' changeability = 'changeable'>
<UML:AssociationEnd.multiplicity> <UML:Multiplicity xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DF0'>
<UML:Multiplicity.range> <UML:MultiplicityRange xmi.id =
'127-0-1-1-1e585d75:116a7befc75:-8000:0000000000000DF1' lower = '1' upper =
'1'/> </UML:Multiplicity.range> </UML:Multiplicity>
</UML:AssociationEnd.multiplicity> <UML:AssociationEnd.participant> <UML:Class
xmi.idref = '-64--88-2-2-109fe0ae:1154067deed:-8000:0000000000000858'/>
</UML:AssociationEnd.participant> </UML:AssociationEnd>
</UML:Association.connection> </UML:Association> </UML:Namespace.ownedElement>
</UML:Model> <UML:Class xmi.id =
'-64--88-18--49--d903a0f:11548fd9b5e:-8000:00000000000009AC' name =
'ArtifactDisplayManager' visibility = 'public' isSpecification = 'false' isRoot
```

= 'false' isLeaf = 'false' isAbstract = 'false' isActive = 'false'/>
<UML:Stereotype xmi.id =
'35-9-22--98-3d03f96e:1154defb87d:-8000:0000000000001041' isSpecification =
'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'>
<UML:Stereotype.baseClass> </UML:Stereotype.baseClass> </UML:Stereotype>
<UML:Package xmi.id = '35-9-22--98-531c0b15:1154e2a0a2b:-8000:0000000000000AD2'
name = 'java' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false'> <UML:Namespace.ownedElement> <UML:Package xmi.id =
'35-9-22--98-531c0b15:1154e2a0a2b:-8000:0000000000000AD3' name = 'lang'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Namespace.ownedElement> <UML:Class xmi.id =
'35-9-22--98-531c0b15:1154e2a0a2b:-8000:0000000000000AD4' name = 'String'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'
isActive = 'false'/> </UML:Namespace.ownedElement> </UML:Package> <UML:Package
xmi.id = '35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AAE' name =
'util' isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract =
'false'> <UML:Namespace.ownedElement> <UML:Class xmi.id =
'35-9-22--98--1199db7d:1154e5689a5:-8000:0000000000000AAF' name = 'Vector'
isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract = 'false'
isActive = 'false'/> </UML:Namespace.ownedElement> </UML:Package>
</UML:Namespace.ownedElement> </UML:Package> <UML:Class xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E57' name = '' visibility
= 'public' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false' isActive = 'false'> <UML:Namespace.ownedElement>
<UML:StateMachine xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E58' isSpecification =
'false'> <UML:StateMachine.context> <UML:Class xmi.idref =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E57'/>
</UML:StateMachine.context> <UML:StateMachine.top> <UML:CompositeState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E59' name = 'top'

isSpecification = 'false' isConcurrent = 'false'> <UML:CompositeState.subvertex>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E5B' isSpecification =
'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E51' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E62' isSpecification =
'false'> <UML:State.doActivity> <UML:CreateAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E5D' name = '[new]'
isSpecification = 'false' isAsynchronous = 'false'>
<UML:CreateAction.instantiation> <UML:Class xmi.idref =
'-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>
</UML:CreateAction.instantiation> </UML:CreateAction> </UML:State.doActivity>
</UML:SimpleState> <UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E65' isSpecification =
'false'> <UML:State.doActivity> <UML:ReturnAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E63' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E6E' isSpecification =
'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E69' name =
'addUserPEnding' isSpecification = 'false' isAsynchronous = 'false'>
<UML:CallAction.operation> <UML:Operation xmi.idref =
'35-9-22--98--28a78339:115482af049:-8000:000000000000089C'/>
</UML:CallAction.operation> </UML:CallAction> </UML:State.doActivity>
</UML:SimpleState> </UML:CompositeState.subvertex> </UML:CompositeState>
</UML:StateMachine.top> </UML:StateMachine> </UML:Namespace.ownedElement>
</UML:Class> <UML:Class xmi.id =

'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E82' name = '' visibility
= 'public' isSpecification = 'false' isRoot = 'false' isLeaf = 'false'
isAbstract = 'false' isActive = 'false'> <UML:Namespace.ownedElement>
<UML:StateMachine xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E83' isSpecification =
'false'> <UML:StateMachine.context> <UML:Class xmi.idref =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E82'/>
</UML:StateMachine.context> <UML:StateMachine.top> <UML:CompositeState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E84' name = 'top'
isSpecification = 'false' isConcurrent = 'false'> <UML:CompositeState.subvertex>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E86' isSpecification =
'false'/> <UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E8C' isSpecification =
'false'> <UML:State.doActivity> <UML:CreateAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E87' isSpecification =
'false' isAsynchronous = 'false'> <UML:CreateAction.instantiation> <UML:Class
xmi.idref = '-64--88-18--49-2af301ad:1153d640b70:-8000:000000000000077E'/>
</UML:CreateAction.instantiation> </UML:CreateAction> </UML:State.doActivity>
</UML:SimpleState> <UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E92' isSpecification =
'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E8D' isSpecification =
'false' isAsynchronous = 'false'> <UML:CallAction.operation> <UML:Operation
xmi.idref = '35-9-22--98--28a78339:115482af049:-8000:000000000000089C'/>
</UML:CallAction.operation> </UML:CallAction> </UML:State.doActivity>
</UML:SimpleState> <UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E95' isSpecification =
'false'> <UML:State.doActivity> <UML:DestroyAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E93' isSpecification =

```
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E98' isSpecification =
'false'> <UML:State.doActivity> <UML:ReturnAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000E96' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
</UML:CompositeState.subvertex> </UML:CompositeState> </UML:StateMachine.top>
</UML:StateMachine> </UML:Namespace.ownedElement> </UML:Class> <UML:Class xmi.id
= '-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EAC' name = ''
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'> <UML:Namespace.ownedElement>
<UML:StateMachine xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EAD' isSpecification =
'false'> <UML:StateMachine.context> <UML:Class xmi.idref =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EAC'/>
</UML:StateMachine.context> <UML:StateMachine.top> <UML:CompositeState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EAE' name = 'top'
isSpecification = 'false' isConcurrent = 'false'> <UML:CompositeState.subvertex>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EB0' isSpecification =
'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EA6' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EB6' isSpecification =
'false'> <UML:State.doActivity> <UML:CreateAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EB1' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EBC' isSpecification =
```

'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EB7' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EBF' isSpecification =
'false'> <UML:State.doActivity> <UML:ReturnAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EBD' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EC2' isSpecification =
'false'> <UML:State.doActivity> <UML:DestroyAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EC0' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EC5' isSpecification =
'false'> <UML:State.doActivity> <UML:DestroyAction xmi.id =
'-64--88-2-2--329cd2b5:115780d7cac:-8000:0000000000000EC3' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
</UML:CompositeState.subvertex> </UML:CompositeState> </UML:StateMachine.top>
</UML:StateMachine> </UML:Namespace.ownedElement> </UML:Class> <UML:Class xmi.id
= '-64--88-2-3-574ca3b5:115c3dafc74:-8000:0000000000000CEB' name = 'Artifact'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'-64--88-2-3-574ca3b5:115c3dafc74:-8000:0000000000000CED' name = 'ArtifactArray'
visibility = 'public' isSpecification = 'false' isRoot = 'false' isLeaf =
'false' isAbstract = 'false' isActive = 'false'/> <UML:Class xmi.id =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D5C' name = '' visibility =
'public' isSpecification = 'false' isRoot = 'false' isLeaf = 'false' isAbstract
= 'false' isActive = 'false'> <UML:Namespace.ownedElement> <UML:StateMachine
xmi.id = '35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D5D'

isSpecification = 'false'> <UML:StateMachine.context> <UML:Class xmi.idref =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D5C'/>
</UML:StateMachine.context> <UML:StateMachine.top> <UML:CompositeState xmi.id =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D5E' name = 'top'
isSpecification = 'false' isConcurrent = 'false'> <UML:CompositeState.subvertex>
<UML:SimpleState xmi.id =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D60' isSpecification =
'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D5A' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D63' isSpecification =
'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D61' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
<UML:SimpleState xmi.id =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D66' isSpecification =
'false'> <UML:State.doActivity> <UML:CallAction xmi.id =
'35-9-22--98-227f752d:115c50ff9c1:-8000:0000000000000D64' isSpecification =
'false' isAsynchronous = 'false'/> </UML:State.doActivity> </UML:SimpleState>
</UML:CompositeState.subvertex> </UML:CompositeState> </UML:StateMachine.top>
</UML:StateMachine> </UML:Namespace.ownedElement> </UML:Class>
</UML:Namespace.ownedElement> </UML:Model> </XMI.content> </XMI>

REFERENCES

REFERENCES

[1] H. Albin-Amiot, P. Cointe, Y.G. Gueheneuc, and N. Jussien. Instantiating and Detecting Design Patterns: Putting Bits and Pieces Together. In *Proceedings of ASE*, 2001.

[2] J. Bansiya and CG Davis. A hierarchical model for object-oriented design quality assessment. *IEEE Transactions on Software Engineering*, 28(1):4–17, 2002.

[3] Berenbach, B., Hartman, J. DesignAdvisor, A UML-based Architectural Design Tool. Siemens Corporate Research, 2002.

[4] Marcel Birkner. Object-Oriented Design Pattern Detection using Static and Dynamic Analysis in Java Software. Master's thesis, University of Applied Sciences Bonn-Rhein-Sieg, Sankt Augustin, Germany, 2007.

[5] L.C. Briand, J.W. Daly, and J. Wüst. A unified framework for cohesion measurement in object-oriented systems. *Empirical Software Engineering*, 3(1):65–117, 1998.

[6] F.P. Brooks. *The mythical man-month: essays on software engineering.* Addison-Wesley, 1995.

[7] S.R. Chidamber and C.F. Kemerer. A metrics suite for object oriented design. *IEEE Transactions on software engineering*, 20(6):476–493, 1994.

[8] J. Clarke, JJ Dolado, M. Harman, R. Hierons, B. Jones, M. Lumkin, B. Mitchell, S. Mancoridis, K. Rees, M. Roper, et al. Reformulating software engineering as a search problem. *IEEE Proceedings - Software*, 150(3):161–175, 2003.

[9] Eclipse.org. Eclipse.org java development environment. `http://eclipse.org/`.

[10] AE Eiben and M. Schoenauer. Evolutionary computing. *Information Processing Letters*, 82(1):1–6, 2002.

[11] R. France, J. Bieman, and B.H.C. Cheng. Repository for model driven development (REMODD). *Lecture Notes in Computer Science*, 4364:311, 2007.

[12] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software.* Addison-Wesley Reading, MA, 1995.

[13] P. Gomes, F.C. Pereira, P. Paiva, N. Seco, P. Carreiro, J. Ferreira, and C. Bento. Selection and Reuse of Software Design Patterns Using CBR and WordNet. In *Proc. of the 15th Int'l Conf. on Software Engineering and Knowledge Engineering (SEKE 2003). San Francisco*, volume 296, 2003.

[14] P. Gomes, F.C. Pereira, P. Paiva, N. Seco, P. Carreiro, J.L. Ferreira, and C. Bento. Using CBR for Automation of Software Design Patterns. *Lecture Notes in Computer Science*, pages 534–548, 2002.

[15] M. Harman and B.F. Jones. Search-based software engineering. *Information and Software Technology*, 43(14):833–839, 2001.

[16] Mark Harman and John A. Clark. Metrics are fitness functions too. In *Proceedings of the 10th IEEE International Symposium on Software Metrics (METRICS '04)*, pages 58–69. IEEE Computer Society, 2004.

[17] C. He, F. He, K. He, J. Liu, and W. Tu. RoleOf Relationship and Its Meta Model for Design Pattern Instantiation. *Lecture Notes in Computer Science*, 3584:642, 2005.

[18] M. Hitz and B. Montazeri. Chidamber and Kemerer's metrics suite: a measurement theory perspective. *IEEE Transactions on Software Engineering*, 22(4):267–271, 1996.

[19] S.U. Jeon, J.S. Lee, and D.H. Bae. An automated refactoring approach to design pattern-based program transformations in java programs. In *Proceedings of the Ninth Asia-Pacific Software Engineering Conference*, pages 337–345. IEEE Computer Society Washington, DC, USA, 2002.

[20] J.R. Koza. *Genetic programming: on the programming of computers by means of natural selection.* The MIT press, 1992.

[21] C. Krämer, C. GmbH, and L. Prechelt. Design recovery by automated search for structural design patterns in object-oriented software. In *Proceedings ofWCRE*, volume 96, page 208, 1996.

[22] S. Marković and T. Baar. Refactoring OCL annotated UML class diagrams. *Software and Systems Modeling*, 7(1):25–47, 2008.

[23] T. Mens. On the use of graph transformations for model refactoring. *Lecture Notes in Computer Science*, 4143:219, 2006.

[24] T. Mens, S. Demeyer, and D. Janssens. Object-oriented refactoring using graph rewriting. Technical report, Technical Report vub-prog-tr-02-01, Vrije Universiteit Brussel, 2001. 213, 2001.

[25] U. Nickel, J. Niere, and A. Zundorf. The FUJABA environment. In *International Conference on Software Engineering*, volume 22, pages 742–745, 2000.

[26] U. Nilsson and J. Małuszyński. *Logic, programming and Prolog.* Wiley, 1990.

[27] M. Ó Cinnéide. *Automated Application of Design Patterns: A Refactoring Approach.* PhD thesis, University of Dublin, Trinity College, 2001.

[28] M. Ó Cinnéide and P. Nixon. A methodology for the automated introduction of design patterns. In *Software Maintenance, 1999.(ICSM'99) Proceedings. IEEE International Conference on*, pages 463–472, 1999.

[29] M. Ó Cinnéide and P. Nixon. Automated software evolution towards design patterns. In *Proceedings of the 4th International Workshop on Principles of Software Evolution*, pages 162–165. ACM New York, NY, USA, 2001.

[30] M. O'Keeffe and M.Ó. Cinnéide. A stochastic approach to automated design improvement. In *Proceedings of the 2nd international conference on Principles and practice of programming in Java*, pages 59–62. Computer Science Press, Inc. New York, NY, USA, 2003.

[31] M. O'Keeffe and M. Ó Cinnéide. Search-based refactoring: an empirical study. *Journal of Software Maintenance and Evolution: Research and Practice*, 20(5), 2008.

[32] M. O'Keeffe and M. Ó Cinnéide. Search-based refactoring for software maintenance. *The Journal of Systems & Software*, 81(4):502–516, 2008.

[33] M.K. O'Keeffe and M. Ó Cinnéide. Getting the most from search-based refactoring. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1114–1120. ACM Press New York, NY, USA, 2007.

[34] W.F. Opdyke. *Refactoring Object-Oriented Frameworks.* PhD thesis, University of Illinois, 1992.

[35] M. OKeeffe and M. Ó Cinnéide. Search-based software maintenance. In *Conference on Software Maintenance and Reengineering (CSMR06)*, pages 249–260, 2006.

[36] J. Paakki, A. Karhinen, J. Gustafsson, L. Nenonen, and A.I. Verkamo. Software metrics by architectural pattern mining. In *Proceedings of the International Conference on Software: Theory and Practice (16th IFIP World Computer Congress)*, pages 325–332, 2000.

[37] R. Poli, WB Langdon, and N.F. McPhee. *A field guide to genetic programming.* Lulu Enterprises Uk Ltd, 2008.

[38] L. Prechelt, B. Unger, W.F. Tichy, P. Brössler, and L.G. Votta. A controlled experiment in maintenance comparing design patterns to simpler solutions. *IEEE Transactions on Software Engineering*, pages 1134–1144, 2001.

[39] B. Schulz, T. Genssler, B. Mohr, W. Zimmer, and F.I. Karlsruhe. On the computer aided introduction of design patterns into object-oriented systems. In *Technology of Object-Oriented Languages, 1998. TOOLS 27. Proceedings*, pages 258–267, 1998.

[40] O. Seng, J. Stammel, and D. Burkhart. Search-based determination of refactorings for improving the class structure of object-oriented systems. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1909–1916. ACM New York, NY, USA, 2006.

[41] N. Shi and R.A. Olsson. Reverse engineering of design patterns from Java source code. *Computer*, 2007.

[42] University of Waterloo Software Architecture Group. SWAG Toolkit. http://swag.uwaterloo.ca, 2008.

[43] G. Sunye, D. Pollet, Y. Le Traon, and J.M. Jezequel. Refactoring UML Models. *Lecture Notes in Computer Science*, pages 134–148, 2001.

[44] L. Tahvildari and K. Kontogiannis. A metric-based approach to enhance design quality through meta-pattern transformations. In *Proc. European Conf. Software Maintenance and Reeng*, pages 183–192, 2003.

[45] Tigris.org. Argouml modeling tool. `http://argouml.tigris.org/`.

[46] M. Torchiano. Documenting Pattern Use in Java Programs. In *Proceedings of ICSM 2002 (International Conference on Software Maintenance)*, pages 230–233, 2002.

[47] L. Torvalds et al. The Linux operating system. `http://www.linux.org/`.

[48] A. Trifu, O. Seng, and T. Genssler. Automated design flaw correction in object-oriented systems. In *Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings. Eighth European Conference on*, pages 174–183, 2004.

[49] N. Tsantalis, A. Chatzigeorgiou, G. Stephanides, and ST Halkidis. Design pattern detection using similarity scoring. *IEEE Transactions on Software Engineering*, 32(11):896–909, 2006.

[50] J.B. Warmer and A.G. Kleppe. *The Object constraint language.* Addison-Wesley, 1998.

[51] W. Zimmer. Relationships between design patterns. *Pattern languages of program design*, 1:345–364, 1995.


MICHIGAN STATE UNIVERSITY LIBRARIES
3 1293 03063 5508