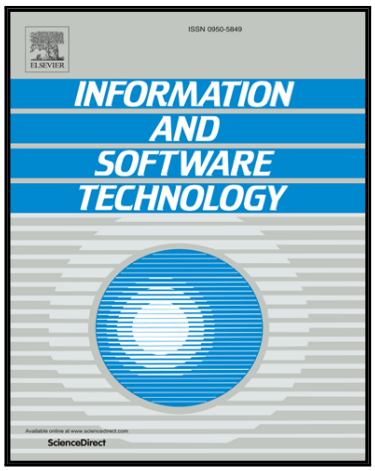
Accepted Manuscript



Fixing Class Design Inconsistencies using Self Regulating Particle Swarm Optimization

Renu George , Philip Samuel

PII: S0950-5849(18)30041-7
DOI: 10.1016/j.infsof.2018.03.005
Reference: INFSOF 5969

To appear in: *Information and Software Technology*

Received date: 2 October 2017
Revised date: 4 March 2018
Accepted date: 5 March 2018

Please cite this article as: Renu George , Philip Samuel , Fixing Class Design Inconsistencies using Self Regulating Particle Swarm Optimization, *Information and Software Technology* (2018), doi: 10.1016/j.infsof.2018.03.005

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Fixing Class Design Inconsistencies using Self Regulating Particle Swarm Optimization

***Renu George*[a,*], *Philip Samuel*[b]**

[a]*Department of Computer Science, Cochin University of Science and Technology, Kochi, Kerala, India*
[b]*Information Technology, SOE, Cochin University of Science and Technology, Kochi, Kerala, India*

ARTICLE INFO

Keywords:
Class diagram,
Activity diagram,
Consistency,
Self regulating particle swarm optimization,
Artificial intelligence

ABSTRACT

Context: The practice of using Unified Modeling Language models during software development and the chances of occurrence of model inconsistencies during software design are increasing. Hence detection of intra-model design inconsistencies is significant in the development of quality software.
Objective: The existing approaches of detecting class attribute inconsistencies rely on human decision making. Manual detection of inconsistencies is exhaustive, time consuming and sometimes incomplete. Therefore, we propose an automated and novel approach to perform consistency check of class attributes using artificial intelligence.
Method: Inconsistency in attribute definition and specification is detected and fixed with Self Regulating Particle Swarm Optimization (SRPSO) algorithm that uses a fitness function to optimize the consistency of attributes in class diagram and activity diagrams. SRPSO is preferred since the best particle is not influenced by its or others experience and uses its direction as the best direction and the remaining particles use self and social knowledge to update their velocity and position.
Result: The use of artificial intelligence technique for detection and fixing of inconsistencies during the software design phase ensures design completeness through generation of models with consistent attribute definitions and a significant improvement in software quality prediction, accurate code generation, meeting time deadlines, and software production and maintenance cost is achieved.
Conclusion: Ensuring consistency and completeness of models is an inevitable aspect in software design and development. The proposed approach automates the process of inconsistency detection and correction in class attribute definition and specification using SRPSO algorithm during the design phase of software development.

1. Introduction

Artificial Intelligence (AI) deals with techniques to make machines intelligent, whereas software engineering deals with the task of designing, implementing and deploying complex systems. Design and implementation of software systems is a much more difficult and challenging engineering discipline since software is well concealed. AI deals with replicating intelligence in machines and hence can be used to solve complex and challenging problems in software engineering. The techniques and algorithms of AI have an impact in almost all areas of software engineering [13].

Model Driven Approach (MDA) proposed by Object Management Group is a design approach for the development of software and the modeling language widely used for requirements modeling and documentation of the system is Unified Modeling Language (UML). UML is a collection of diagram centric design notations that are loosely connected. UML provides a facility to design a software system by expressing independent views. Class diagrams and activity diagrams portray the static and dynamic views of the system being designed. The implementation of a class diagram method, represented with an activity diagram involves class attributes. Class design inconsistency arises if the attributes referenced in the activity diagram is undefined in the class.

Large complex systems require different models. A model, representing the system, is group of UML model elements at a given stage of abstraction. Inconsistencies occur between UML models and requirements, standards, or other design artifacts or within a UML model. Modification or improvement in the models may also lead to inconsistencies. Inconsistency may occur due to constraint violation and contradictory requirements. The chances of making contradictory decisions are more and detection of inconsistencies has become vital for the development of accurate software. Inconsistencies result directly in software defects. The defects can be prevented by systematic UML model consistency checking. Two types of consistency are defined on design models: horizontal or intra-model consistency and vertical or inter-model consistency. The consistency possessed by different model elements belonging to the same version is named as intra-model consistency. The consistency expected between a model and its different versions is named as inter-model consistency. Model elements of the same version may represent the static view as well as the dynamic view of the modeling domain [3].

Particle swarm optimization (PSO) introduced in 1995 by Kennedy and Eberhart is motivated by the cooperative behavior of a flock of birds or a school of fish seeking for food. PSO is an optimization technique based on population with computational intelligence [1]. The concept of PSO is simple, uses primitive operators, searching mechanism is unique, is easy to implement, and is computationally efficient and inexpensive. The properties of PSO

* *Corresponding author.*
E-mail address: renugeorge@ceconline.edu (Renu George)

ACCEPTED MANUSCRIPT

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
✓ امکان دانلود نسخه ترجمه شده مقالات
✓ پذیرش سفارش ترجمه تخصصی
✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
✓ دانلود فوری مقاله پس از پرداخت آنلاین
✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات