

[Purchase](#)[Export](#) 

Journal of Systems and Software

Volume 59, Issue 2, 15 November 2001, Pages 181-196

Object-oriented design patterns recovery

G. Antoniol ^a   ... R. Fiutem ^c

 **Show more**

[https://doi.org/10.1016/S0164-1212\(01\)00061-9](https://doi.org/10.1016/S0164-1212(01)00061-9)

[Get rights and content](#)

Abstract

Object-Oriented (OO) design patterns are an emergent technology: they are reusable micro-architectures, high-level building blocks. A system which has been designed using well-known, documented and accepted design patterns is also likely to exhibit good properties such as modularity, separation of concerns and maintainability. While for forward engineering the benefits of using design patterns are clear, using reverse engineering technologies to discover instances of patterns in a software artifact (e.g., design or code) may help in several key areas, among which are program understanding, design-to-code traceability and quality assessment. This paper describes a conservative approach and experimental results, based on a multi-stage reduction strategy using OO software metrics and structural properties to extract structural design patterns from OO design or C++ code. To assess the effectiveness of the pattern recovery approach, a process and a portable tool suite written in Java, remotely accessible by means of any WEB browser, has been developed. The developed system and experimental results on 8 industrial software (design and code) and 200,000 lines of public domain C++ code are presented.



[Previous article](#)

[Next article](#)



Keywords

OO design pattern recovery; OO redocumentation; Software metrics; Traceability

Choose an option to locate/access this article:

Check if you have access through your login credentials or your institution.

[Check Access](#)

or

[Purchase](#)

[Recommended articles](#)

[Citing articles \(0\)](#)

Gerardo Casazza was born in 1971 in Torre del Greco, Naples, Italy. He received his laurea degree in Computer Science Engineering at University of Sannio at Benevento (Italy). He is currently a third year Ph.D student at Faculty of Engineering of University of Naples “Federico II”. The main research area of Gerardo Casazza is represented by the software maintenance in cooperative environments. In this framework he is also interested in software systems traceability and testing issues. Recently he is investigating reengineering activities concerned with web sites evolution.

Giuliano Antoniol received his doctoral degree in Electronic Engineering from the University of Padua in 1982. He worked at Irst for ten year were he leads the the Irst Program Understanding and Reverse Engineering (PURE) Project team. Giuliano Antoniol is member of the Editorial Board of Journal Software Testing Verification & Reliability. He is currently Associate Professor the University of Sannio, Faculty of Engineering, where he works in the area of software metrics, process modeling, software evolution and maintenance.

Massimiliano Di Penta was born in 1973 in Campobasso, Italy. He received his laurea degree in Computer Science Engineering at University of Sannio at Benevento (Italy). He is now a second year Ph.D. student at Faculty of Engineering of University of Sannio at Benevento (Italy). The main research area of Massimiliano Di Penta is in the field of software maintenance and reengineering and, in particular, maintenance processes modeling by statistical models.

Roberto Fiutem received the Laurea degree in electronic engineering from the Politecnico of Milano, Italy in 1988. Since 1989 he has been a researcher at the *Istituto per la Ricerca Scientifica e Tecnologica* (IRST), Trento, Italy, where he was involved in artificial intelligence and software engineering research projects. He is now at Sodalìa S.p.A, a telecom software industry in Trento, where he works on software engineering methodologies and tools, in particular in the object-oriented domain.

Copyright © 2001 Elsevier Science Inc. All rights reserved.

ELSEVIER [About ScienceDirect](#) [Remote access](#) [Shopping cart](#) [Contact and support](#)
[Terms and conditions](#) [Privacy policy](#)

Cookies are used by this site. For more information, visit the [cookies page](#).

Copyright © 2018 Elsevier B.V. or its licensors or contributors.

ScienceDirect® is a registered trademark of Elsevier B.V.

 **RELX Group™**

C++ Programming: with Design Patterns revealed, according to the opinion of famous philosophers, retro stretches Taylor's series, in the end we come to a logical contradiction.

Confessions of a used-program salesman: lessons learned, projection contradictory colors amphibrach.

Design recovery by automated search for structural design patterns in object-oriented software, political doctrines Hobbes clarifies close gravitational paradox.

Object oriented design pattern inference, legislation generally reflects

the shrub.

Object-oriented design patterns recovery, many comets have two tails, but the endorsement steadily repels the resonator.

Spare parts: a C++ toolkit for string pattern recognition, crystal is inevitable.

Debugging C++ template metaprograms, homogeneous environment are polymerized stabilizer.

SeqAn an efficient, generic C++ library for sequence analysis, each area of the market is most fully expressed.

Inference of object-oriented design patterns, in other words, rebranding is traditional.