Advances in Engineering Software 42 (2011) 12–24

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Object-oriented programming paradigm for damage tolerant evaluation of engineering structural components

A. Rama Chandra Murthy *, G.S. Palani, Nagesh R. Iyer

CSIR-Structural Engineering Research Centre, CSIR Campus, Taramani, Chennai 600 113, India

ARTICLE INFO

Article history:
Received 7 January 2008
Received in revised form 7 January 2010
Accepted 12 October 2010
Available online 12 November 2010

Keywords:
Damage tolerant evaluation
Fracture mechanics
Object-oriented programming
Crack growth
Remaining life
Residual strength

ABSTRACT

This paper proposes a new fracture mechanics based OOP tool for damage tolerant evaluation of cracked structural components including tubular joints subjected to constant and variable amplitude loading. To meet requirements of damage tolerant evaluation of structural components, interactive and user-friendly software, has been developed by using OOP concepts. Application of OOP concepts with class and sequence diagrams generated using unified modified language (UML) design tool has been explained with reference to the software. Graphical user interface (GUI) has been developed using VC++, which acts as a client at the front end, while the database developed using MS-ACCESS-XP acts as the server at the back-end. Database design for typical structural components with different crack configurations has been shown in the form of tables. The details of various program modules and structure of GUI have been outlined. Number of benchmark problems has been solved through GUI for verification and validation. The efficacy of the software has been illustrated through an example problem.

1. Introduction

The advent of high-speed computers fostered the inception of a number of computer programs for analysis and design of civil engineering structures. FORTRAN has traditionally been the popular programming language for writing scientific codes. Later, numerous revisions took place to incorporate more analysis options, enhanced computational capabilities and the latest knowledge based on a single computational platform. Most of the computer programs employ the conventional waterfall algorithm-driven procedural programming approach, lacking from maintenance, management and expansion of large scale computer codes [1]. Object-oriented programming (OOP) is a programming technique that was developed with the objective of addressing some of the limitations of procedural programming approach and has become the order of the day in programming. OOP concepts and practices have been brought together, with associated terminology, to create a new programming framework. Together, the ideas behind OOP are said to be so powerful that they create a paradigm shift in programming. OOP has tremendous potential in the development of scientific software as it promises to produce more and more easily maintainable software. A major benefit of OOP is that the interface with a particular architecture can take place at the library-code level, where the machine dependent details of data storage, synchronization and parallelism are implemented in lower-level objects, and not at the user code level, which deals with the high level objects associated with the mathematics of the physical problem. One of the powerful features of OOP is the capability of sharing attributes and skills. The most important point, however, is that those issues directly addressed by OOP software development, i.e., maintainability and usability are becoming dominant.

Zimmermann et al. [2] demonstrated the usefulness of this alterative programming paradigm in the field of finite elements. The basic concept of the object-oriented approach was discussed in the context of the implementation of a simple finite element program using Smalltalk language. Dubois-Pelerin et al. [3] used the prototyping language Smalltalk for development of object-oriented implementation of the finite element method. Emphasis was laid on the potential of the concepts of object-oriented programming, mainly data encapsulation and inheritance. The OOP paradigm with its characteristics of abstraction, inheritance modularity and encapsulation of data and operations provides a highly flexible and modular programming environment for analysis and design of complex software [4–5]. Dubois-Prlerin and Zimmermann [6] showed that performances comparable with FORTRAN can be achieved, without sacrificing too much of the qualities of the prototype program: code understandability, code extendability and code debugging. As OOP provides enormous potential in the development of scientific software, there exists enormous scope for OOP in structural engineering applications. Eyheramendy and

* Corresponding author. Tel.: +91 44 22549205; fax: +91 44 22541508.
E-mail addresses: murthyarc@sercm.org, archandum@yahoo.com (A.R.C. Murthy).

0965-9978/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2010.10.012

Zimmermann [7] applied the principles of object-oriented programming directly to the problem statement in differential form. All derivations were carried out in symbolic form except for that of the final numerical matrix form. Salgado and Aliabadi [8] analysed the cracked stiffened panels by using dual boundary element method. Besson and Foerch [9] discussed the aspects of object-oriented finite element design which become relevant as the project size increases. A simple core library was described which aids program development by isolating repetitive tasks into optimized classes. Salgado and Aliabadi [10] developed an interactive, integrated system for damage tolerance design of aircraft stiffened panels. The structural configuration was represented in an object oriented fashion, using a high level of abstraction and in terms of object components well-known to the designer, such as stiffeners, cracks and boundaries. Crack propagation analysis was performed with the dual boundary element method which allows for the solution of problems involving single or multiple cracks under mixed mode conditions In 1990s, research efforts were directed towards developing object oriented software architectures for computational techniques such as finite element methods [5]. Significant research was carried out in the area of object-oriented finite element modeling and analysis for the solution of complex engineering problems [4,5,11–15]. Commend and Zimmermann [16] developed object-oriented finite element program for static and dynamic nonlinear applications. von Mises plasticity including isotropic and kinematic hardening was described in detail. Assaf-Al and Saffrini [17] used OOP concepts for optimization of concrete slabs by combining modeling, structural analysis, concrete design, cost estimation and optimization. Cali and Citarella [18] evaluated the residual strength for a cracked butt-joint based on *R*-curve analysis and plastic collapse condition. Ugwu et al. [19] highlighted the application of an object oriented frame work to decision making in designing for durability in the bridge domain. Bordas and Moran [20] described coupling of X-FEM with level set methods that can be used to solve complex three dimensional industrial fracture mechanics problems through combination of an object-oriented (C++) research code and a commercial solid modeling/ finite element package (EDS-PLM/I-DEAS). Qiao [21] presented the implementation aspects of object-oriented programming approach for the boundary element method in two dimensional heat transfer analysis. Fang et al. [22] developed an object oriented frame work with interactive graphics for pavement studies using finite element analysis.

Despite several intensive research studies undertaken with the objective of developing object oriented software for structural engineering applications, the implementation of OOP concepts for advanced applications such as damage tolerant evaluation of engineering structural components are only a few and limited to damage tolerance design of stiffened panels [8,10]. Therefore, a need has been felt to develop interactive and user-friendly software employing OOP concepts which will aid the analysts and designers with fracture mechanics based OOP tool for damage tolerant evaluation of cracked structural components including tubular joints subjected to constant and variable amplitude loading.

It is well-known that damage tolerant evaluation of structural components has two major objectives, namely, evaluation of crack growth life and residual strength evaluation. In this paper, methodologies are presented for assessment of either crack growth life or residual strength or both for structural components subjected to either constant or variable amplitude loading. In the case of variable amplitude loading, the famous models such as Wheeler, Willenborg and improved Wheeler are used to account for retardation effects. The proposed improved Wheeler model in the paper makes use of expressions for shaping exponent. The improved Wheeler model has also been extended to account for tensile–compressive overloads. In this model, plastic zone size is modified appropriately to account for compressive overload. Further, remaining life approach proposed for residual strength evaluation is the more rational and reliable compared to the generally used plastic collapse condition or fracture toughness criterion. The focus of the paper is that all these concepts of damage tolerant evaluation of structural components subjected to constant and variable amplitude loading have been brought under the object oriented paradigm. The necessary objects, classes and data structures have been developed and discussed in the paper. Application of OOP concepts with class and sequence diagrams generated using unified modified language design tool has been explained with reference to the software. Graphical user interface (GUI) has been developed using VC++, which acts as a client at the front end, while the database through MS-ACCESS-XP acts as the server at the back-end. Database design for typical structural components with different crack configurations has been shown in the form of tables. The details of various program modules and structure of GUI have been outlined. Number of benchmark problems has been solved through GUI for verification and validation. The efficacy of the software has been illustrated with one example problem.

2. Intrinsic worth of object-oriented programming modeling for analysis of structures

The idea behind OOP is that a computer program may be seen as comprising a collection of individual units, or objects, that act on each other, as opposed to a traditional view in which a program may be seen as a collection of functions, or simply as a list of instructions to the computer. Each object is capable of receiving messages, processing data, and sending messages to other objects. Each object can be viewed as an independent little machine or actor with a distinct role or responsibility. OOP is claimed to promote greater flexibility, maintainability and reusability and is widely popular in large-scale engineering software. OOP is based on a collection of discrete objects that incorporate data structures and behavior. Each data structure has, combined with it, the procedures which apply to that data structure. These entities called objects can be associated to one another in one network, rather than two [23,24]. It contrasts with conventional programming, in which data structures and behavior are only loosely connected. OOP is often called a paradigm rather than a style or type of programming, to emphasize the point that OOP can change the way software is developed by actually changing the way in which programmers and software engineers think about software development. The main OOP features include, data abstraction, data encapsulation, inheritance and Information sharing, etc., whereas benefits of OOP include code reuse, ease of maintenance and enhancement, fewer and shorter iterations, etc.

3. Damage tolerant evaluation

Reliability and functionality are two of the most important requirements of engineered structures and components. Most of the structures such as offshore structures, nuclear containments, reactor vessels, flyovers, high-rise buildings, aerospace structures, ship hulls, bridges and components are required to operate under stringent operating conditions. The environment may also be variable, regardless of the operating regime. All the design calculations are made with the fundamental assumption that there is no damage or flaw in the structural components or in the structure. Nevertheless in reality not all engineering components or elements used in the structures are flawless. They are either damaged or imperfect during manufacturing or while in service. These damages restrain the usage of the structure for the intended purpose