A Prototype of a Reengineering Tool for Transforming Imperative Programs into Object-Oriented Specifications

Bian Zhangsheng, Bao Dongyi
E-mail: 8812384@mail.dyu.edu.tw

ABSTRACT
Software reengineering from legacy systems has become a growing concern as more and more organizations perceive their legacy systems as valuable assets and the competitive edge in business process reengineering. The purpose of software reengineering is to transform legacy systems into counterparts based on a new technology so that the software would gain an improvement in reusability and maintainability. Compared with structured techniques, the object technology can produce more flexible software and reduce the cost of maintainance. This paper presents the design of a software reengineering tool that supports the transformation from logical imperative programs with database schema into object-oriented specifications and attempts to promote reusability and maintainability of the software by applying object-oriented methodology and object-oriented design patterns. Moreover, from the perspective of input-process-output (IPO) information processing model including data flow diagrams we invented the IPO design pattern and adopted it in the design of architecture of this tool. We expect that in the future this reengineering tool would be combined with other research results in different abstraction levels to form a complete CASE environment which will benefit business organizations in process reengineering and strategic planning.

Keywords: software reengineering; legacy systems; object technology; object-oriented design patterns

Table of Contents

第一章 緒論
第一節 研究背景
第二節 研究動機
第三節 研究目的
第四節 研究範圍
第五節 研究方法
第六節 論文架構

第二章 文獻探討
第一節 物件導向的基本概念
第二節 物件導向分析
第三節 物件導向設計
第四節 物件導向軟體重複使用的方式
第五節 物件導向設計樣式
第六節 軟體再工程技術與工具

第三章 系統架構分析與設計
第一節 轉換程序概觀
第二節 啟發式法則與演算法
第三節 再工程工具之系統架構
第四節 系統設計樣式

第四章 系統雛型實作
第一節 Module_Allocator子系統模型
第二節 演算法之實作
第三節 Module_Allocator子系統之圖形使用者介面

第五章 結論與後續研究建議
第一節 結論
第二節 後續研究建議

REFERENCES

[1] 張海藩,「軟體工程導論」,格致,1993年。
[21] Jacobson, I. et al., Object-Oriented Software Engineering. Readings,