An Application of Artificial Intelligence to Prototyping Process in Performance Design for Real-Time Systems

Shinichi Honiden¹ Naoshi Uchihira¹ Kiyoshi Itoh²

1 Systems and Software Engineering Laboratory, Toshiba Corporation
70 Yanagi-cho, Saiwai-ku, Kawasaki 210, Japan
honiden@ssel.toshiba.co.jp

2 Faculty of Science and Technology, Sophia University
7-1. Kioi-cho, Chiyoda-ku, Tokyo 102, Japan
itohkiyo@hoffman.cc.sophia.ac.jp

ABSTRACT

This paper describes an application of artificial intelligence technology to the implementation of a rapid prototyping method in Object-Oriented Performance Design (OOPD) for real-time systems. A prototyping process is composed of three steps: Prototype construction, Prototype execution, and Prototype evaluation. The authors present the following artificial intelligence based methods and tools to be applied to each step. In the prototype construction step, a rapid construction mechanism, using reusable software components, is implemented based on the planning method. In the prototype execution step, a hybrid inference mechanism is used to execute the constructed prototype which is described in declarative knowledge representation. In the prototype evaluation step, an expert system, which is based on qualitative reasoning, is implemented to detect and diagnose bottlenecks and generate an improvement plan for them.
1. INTRODUCTION

Object-oriented technology can provide designers with practical, productive ways to develop software in most of application areas. As for real-time applications, object-oriented technology has been practically employed in a number of systems. Recently, various multiprocessors have become commercially available and have been used in many applications. Since stringent performance requirements are inevitable for real-time applications, it is very important to predict the target system performance precisely during the design phase, in order to determine the optimal software and hardware configurations which will satisfy users' requirements. However, in some real-time systems on multiprocessors, it is quite difficult to adjust some of the performance factors, such as load-balance on a given multiprocessor architecture, to satisfy the performance requirements. In practice, the performance design activities in the object-oriented design (called Object-Oriented Performance Design: OOPD) process tend to be empirical, because there are few algorithms which can derive the optimal software configurations that satisfy the performance requirements under a given hardware configuration. In other words, OOPD activities are usually carried out on a trial-and-error basis. These activities may compose a prototyping approach to improving ill-defined problems where only few algorithms are available. Artificial intelligence technology has been effectively used to tackle ill-defined problems. Various ill-defined problems are actually seen in the software engineering field including programming tasks. Several attempts have been made in applying the artificial intelligence techniques to these ill-defined problems in the software engineering field. Typical ones are GIST [Coh84, Swa83], Programmer's Apprentice [Ric78], φ0 [Bar82], Glitter [Fic85], Data-Representation Advisor [Kat81], SC [Dow90], and AFFIRM [Ger80]. These methods seem to be successfully applied to the particular phases or domains. Individual methods can be used to handle several steps in the software development process, but none of them can cover the overall development process nor can be used to fully implement the prototyping process.

A prototyping process is defined to be composed of three steps: prototype construction, prototype execution, and prototype evaluation [Ito89a]. The authors present the following artificial intelligence based methods and tools to be used in these three steps.
1) In the prototype construction step, a rapid construction mechanism, using reusable software components, is implemented, based on the planning method.
2) In the prototype execution step, a hybrid inference mechanism is used to execute the prototype which is described in declarative knowledge representation.
3) In the prototype evaluation step, an expert system, which is based on qualitative reasoning, is implemented to detect and diagnose bottlenecks and generate appropriate improvement plans.