

Partitioning and Allocation of Objects in Distributed Application Development

G.C. Low

School of Information Systems
The University of New South Wales
Sydney, NSW 2052 Australia
email: G.Low@unsw.edu.au

G. Rasmussen

Currently at Poynton and Partners, Perth, Australia

In recent years there has been significant interest in the use of object-oriented techniques for the production of distributed systems. However, only a limited amount of work has been reported on the incorporation of distributed system design issues into object-oriented development methodologies. The identification of processes within a software system (partitioning) and the allocation of these processes to processors in the system is a fundamental problem in the design of distributed systems.

Techniques for evaluating the three objectives of task partitioning: minimizing inter-module communications, exploiting concurrency, and limiting the size of processes are presented. The potential concurrency in the object model is graded into three categories. This grading of the potential concurrency between objects/classes is new and is anticipated may assist in the partitioning decision making process. Communication and execution costs required for software allocation are determined from the event diagrams and event traces. The aim of the paper is to use the object-oriented model to produce information that fits into existing non object-oriented decision techniques such as traditional graph theoretic or graph heuristic allocation techniques.

CR Categories: Object-oriented design methods and software engineering practices; distributed systems.

1. INTRODUCTION

In recent years there has been significant interest in the use of object-oriented techniques for the production of distributed systems. However, the majority of work has been at the implementation level. Several distributed object-oriented programming languages (Black *et al*, 1987) have been developed, as well as distributed object-oriented operating systems (Bennett, 1987) and systems to provide distribution services to object-oriented applications (Atkinson, 1991). Only a limited amount of work has been reported on the incorporation of distributed system design issues into object-oriented development methodologies. Low *et al*, (1996) discussed the modifications required, in general terms, for a generic OO development methodology. Rasmussen *et al*, (1996a) applied this work to the MOSES methodology.

Copyright© 2000, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: May 2000

Guest Editor: Brian Henderson-Sellers