Manuscript: OPEN/REENGINE.TEX

OPEN Re-engineering

B. Henderson-Sellers

School of Information Technology

Swinburne University of Technology,

Hawthorn, Victoria, Australia

Printed: July 28, 1997

[submitted to *ESEC/FSE '97 Workshop on Object-Oriented Re-engineering*

Zurich, Friday September 26, 1997]

Address for Correspondence

Professor B. Henderson-Sellers
Director, Centre for Object Technology Applications and Research
School of Information Technology
Swinburne University of Technology
John Street
PO Box 218
Hawthorn
Victoria 3122
AUSTRALIA
fax: +61 3 9819 0823
email: brian@csse.swin.edu.au

OPEN Re-engineering

Position Paper

by B. Henderson-Sellers

for ESEC/FSE '97 Workshop on Object-Oriented Re-engineering

Abstract

The third generation OO methodology OPEN (Object-oriented Process, Environment and Notation) encompasses the full lifecycle of software development. It is structured in terms of a fully objectified lifecycle together with a set of Tasks and Techniques. Here, those Tasks and Techniques pertinent to re-engineering are briefly discussed together with evaluation of their relevance to migration issues (from a traditional information system to an object-oriented architecture).

1. OPEN outline

OPEN (Object-oriented Process, Environment and Notation) is a third generation process-focussed OO methodology (Graham *et al.*, 1997) which is built around a lifecycle metamodel defining its software engineering process architecture (SEPA). Each instantiation of the SEPA is an individual SEP (software engineering process) tailored from and compatible with the overall architecture of OPEN. A single SEP is tailored to be highly appropriate to a particular problem type and industry domain. Each SEP is composed of objects representing Activities. To permit this tailoring, Activities are connected in a flexible fashion by stating preconditions that must be satisfied before an Activity can commence. The "methods" of an Activity are its Tasks. Tasks can also be considered the smallest unit of work within OPEN. Tasks are carried out by agents (people) using Techniques. A two-dimensional matrix links the Task (which provides the statement of goals i.e. the 'what') to the Techniques (which provide the way the goal can be achieved i.e. the 'how'). Techniques range across project management, inspections and so on through to detailed theories and practices for requirements engineering and system modelling. OPEN provides a large repository of Techniques taken from and supplementing existing methods.

2. Migration and Re-engineering

Organizations wishing to "modernize" their information systems into the world of object-orientation may encounter two issues, often linked: (organizational) migration and (system) re-engineering. Migration relates to the way in which individuals and their organization learn about the techniques of object technology. Secondly, at the technical level, existing, so-called legacy, systems may need to be changed in such a way (re-engineered) as to be compatible with the OO architecture of newly built systems.

The former issue (migration) can be solved, *inter alia*, by strategic and operational planning and by a well thought out education and training programme. The latter problem of ensuring interoperability between existing (e.g. COBOL) and new (e.g. Java) systems is often less easily solved since the knowledge of the detailed working of a large COBOL system frequently resides in the brain of an individual. Indeed, it is often the demise of that individual that triggers the idea of re-engineering (Swatman, p.c., 1997).

3. OPEN Support for Migration and Re-engineering

Both by analyzing the literature and undertaking collaborative studies with industry of migration strategies, we are currently working to enhance existing OPEN support for migration and re-engineering. The need for such a set of well-defined heuristics is underlined by Baudoin and Hollowell (1996, p408). At present, OPEN includes Techniques such as Wrappers and Tasks such as Evaluate Quality and Integrate with Existing Non-OO Systems within its methodological framework — derived in part from the earlier work in MOSES (Henderson-Sellers and Edwards, 1994) and SOMA (Graham, 1995). These can be described in more detail in the Workshop.

4. Summary

Methodological issues for re-engineering are twofold: support for the analysis/design/code Tasks and Techniques encountered when considered how a legacy system can be re-engineered into a fully OO system (as outlined here in the context of the OPEN methodological framework) and support in terms of guidelines, heuristics and (possibly) a "blueprint" on how an organization could best follow a successful migration path from non-OO to an OO development environment. Currently, migration strategies, whilst touched upon in MOSES, need further developmental work within the context of an OO methodology to be of optimum assistance to industry in their move to an OO future.

5. Acknowledgements

I wish to thank Ian Graham for his continuing collaboration and Paul Swatman for his insightful comments. This is Contribution no 97/26 of the Centre for Object Technology Applications and Research (COTAR).

References

Baudoin, C. and Hollowell, G., 1996, *Realizing the Object-Oriented Lifecycle*, Prentice Hall, New Jersey, USA, 508pp

Graham, I.M., 1995, *Migrating to Object Technology*, Addison–Wesley, Wokingham, 552pp

Graham, I., Henderson-Sellers, B. and Younessi, H., 1997, *The OPEN Process Specification*, Addison-Wesley, UK

Henderson-Sellers, B. and Edwards, J.M., 1994, *BOOKTWO of Object-Oriented Knowledge: The Working Object*, Prentice-Hall, Sydney, Australia, 594pp