# Reducing Cognitive Distance:
# The Role of an Effective Reuse Program

Peter J. Bell

Australian Centre for Unisys Software
115 Wicks Road, North Ryde
Sydney. NSW, Australia 2113.
Tel: (61 2 3901361)
Email: peter@syacus.acus.oz.au

**Abstract**

This position paper states that the goal of an effective reuse program is to reduce "cognitive distance", where this is defined as the time between initial concept and final implementation. It presents the barriers to introducing a reuse program into an organisation as a result of ongoing experiences in such a program in an Object-Oriented development environment. The attitudes of both developers and management are presented as two of the main barriers, and the methods used to address these barriers are discussed. The requirement to focus on well-specified, robust and efficient assets is identified. Practical side benefits of introducing a reuse program are also highlighted.

**Keywords:** Reuse, frameworks, reuse barriers, reuse guidelines

**Workshop Goals:** Learning: Networking: Understanding the state of the practice.

**Working Groups:** Reuse Management, Organisation and Economics, Domain Analysis / Engineering, Design Guidelines for Reuse & C++, Reuse and 00 methods, Useful and Collectible Metrics, Reuse Handbook

# 1 Background

The Australian Centre for Unisys Software (ACUS) is the primary centre for research and development activities for Unisys in Australia. It was created in 1987 and has since established itself as a software centre specialising in the development of distributed applications and graphical user interfaces. Object oriented languages have been used at ACUS since 1988.

In October of last year (1992), the position of 'reusability engineer' was created to ensure that the benefits of reuse that object oriented languages claim to support were being realised in the organisation. The roles of the reusability engineer are primarily aimed at establishing a culture of reuse within ACUS and to develop a library of in-house and third party reusable objects (designs, code, test harnesses, etc). These roles are specified in more detail in the following sections.

## 1.1 Creation and Maintenance of Reusable objects

This involves finding reusable objects in existing projects and modifying (where necessary) to make the object satisfy a more generic set of requirements. A set of reuse guidelines have been written in order to aid project developers in designing with reuse in mind.

This role also involves the ongoing evaluation of third party libraries to determine suitability for project requirements.

## 1.2 Involvement in Project Design

This role is intended to promote the re-use of the library objects and frameworks in new project designs and to ensure that new software components are designed with reusability in mind. The reusability engineer has the responsibility of ensuring that projects both make use of existing reusable artifices and contribute to the ongoing development of the library of reusable items.

## 1.3 Alterations to Existing Development Processes:

The definitions of most of the development processes used within ACUS have been (or are being) altered to cater specifically for reusability. The intention is that awareness of reuse issues will be greater if they are made a part of the development processes used within the organisation.

## 1.4 Development of Reusable Frameworks

The development of reusable frameworks takes reuse beyond isolated objects. A collection of interrelated objects when combined into a useful framework can dramatically reduce the development time for applications which make use of such frameworks. One of the more challenging roles of the reusability engineer is to establish what frameworks exist within the organisation and to ensure that they are developed in a manner that allows the framework to be utilised by future projects. It also requires that new frameworks be developed as a result of careful domain analysis.

### 1.5 Corporate Strategy Development

Several development plants within Unisys are now combining their reuse experiences in an object oriented environment in order to establish a corporate strategy towards reuse. ACUS, is contributing to this strategy development.

### 1.6 Research Activities and University Contacts

Given that reuse is an area of significant research, the reusability engineer is also responsible for staying up to date with research activities and establishing contacts with the relevant researchers both locally and overseas.

## 2 Position - Narrowing the Cognitive Distance

If the cognitive distance is the effort between the initial concept and final implementation, then a major obstacle for effective reuse within an organisation is the difficulty in creating and developing an attitude that aims at reducing the cognitive distance. This is true for both developers and management. Developers have to believe they can find and modify (where necessary) a reusable object that suits their needs faster than they can build it themselves. Similarly management must be educated to understand where the long term savings of establishing a reuse program will come from. Our goal is to raise the perception that reuse activities in the development cycle are not only beneficial but critical to the long term viability of a commercial software organisation.

### 2.1 Changing Developer's Attitudes

The need to change the developers attitudes towards reuse stems from the 'not invented here syndrome'. In the past the perception has been that the effort to understand, modify and incorporate someone else's code into your own was too great. Unfortunately the perception has too often been an accurate one.

The role of the reusability engineer is to work with the developers and ensure that they have at their fingertips the information they need to quickly and easily assess the suitability of any given reusable object. The phrase "well- specified correct, robust and efficient" [1] has become the key to turning developers attitudes around. Our experience has been that if they can understand easily what the initial requirements of an object were, then they can assess whether it matches their needs. Similarly a common regression test procedure allows the developer to determine an objects stability and to see if a particular test case is catered for and able to be handled by the object. If these two criteria are applied to all reusable objects, i.e. developers know what it is supposed to do and that it is done correctly, then they will happily reuse someone else's code.

### 2.2 Changing Management's Attitudes

The second aspect of changing developers attitudes deals with new reusable designs and code. Whilst achieving reuse of existing objects is half the battle, the library needs to grow in order to remain viable. This brings the developers into conflict with managers. Until managers perceive a real benefit from reusing designs on code, they will remain very reluctant to allow more time in project schedules than is necessary to get the job done. This leaves little room for the work required to generalise a good design into a reusable design.

Our expectation is that only when we are in a position to require projects to submit a certain percentage of their development work for inclusion in the library, will a change in attitude come about. In the interim, it is the responsibility of the reusability engineer to solicit new reusable objects from the project teams. Involvement in design discussions enable more generic designs to be achieved. Again it is a gradual change in attitude that will bring about effective reuse.

### 2.3 Beneficial Side Effects

Given that the key to the reuse program is "well specified correct, robust and efficient" designs and code, several safeguards have been put in place which stand to have beneficial side-effects for project teams.

The entrance criteria are strict and uncompromising. This leads to a development approach for reusable code that is more exacting than would otherwise be necessary. If this feeds back into the project teams, the results can only be beneficial to the quality of delivered products. The level of documentation, the use of standard testing mechanisms and adherence to programming guidelines are all strongly monitored for objects in the reuse library. This is necessary in order to ensure consistency within the library. Only by maintaining the consistency and stringent requirements on quality can we hope to achieve effective reuse - to narrow the cognitive distance. If this insistence on quality improves the processes used within the project teams, then it becomes a beneficial side effect.

## 3 Comparison

### 3.1 Experience Report on Software Reuse Project

In the proceedings of the 14th annual conference on software engineering, Sadahiro Isoda [2] reported on his experiences with specific software reuse projects. The scale of his efforts was significantly bigger than our own, particularly in terms of the resources that were available to him. However, his findings matched our perceptions quite closely. Of paramount importance was the support of senior management. Only with this support can a reuse program hope to survive. Also of interest to us was the establishment of a reusability committee to oversee the activities of the reuse program. We have attempted to emulate this by ensuring that the reusability engineer works closely with each of the project groups. One thing we have taken notice of is the suggestion that a paper based searching mechanism will suffice for the library up until it contains about 1000 objects. This reduces our need to spend time on developing/acquiring support tools and allows us to concentrate on developing the content of the library itself.

## References

[1] M. Cline and D. Lea, *Using Annotated C++*. Clarkson University and SUNY, Owego, 1991.

[2] S. Isoda, "Experience report on software reuse project: Its structure activities and statistical results," in *Proceedings of 14th International Conference on Software Engineering*, 1992.

# 4 Biography

Peter L Bell is a senior software engineer with the Australian Centre for Unisys Software (ACUS), Sydney, Australia. He holds the position of Reusability Engineer and is responsible for the development of reuse activities within ACUS. This includes the development of a reuse library for multiple platforms and the establishment of processes which will result in more reusable designs. He is working, in conjunction with others in Unisys to establish a corporate strategy towards reuse. Prior to taking up this position, he worked on several software development projects in the areas of distributed computing and graphical user interfaces using object oriented languages. He received a B.E. (Elec) from the University of New South Wales in 1987.