System Development in the New Millennium: An Australian Perspective on OO Adoption

Ee Kuan Low*
Angèle L M Cavaye**
Aileen Cater-Steel*

*Department of Information Systems
University of Southern Queensland
Queensland, Australia
Email: eekuanlow@usq.edu.au

**Graduate School of Management
Southern Cross University
New South Wales, Australia

Abstract

In an increasingly turbulent business environment, organisations increasingly rely on properly understood and effectively managed systems development methodologies (SDM) as the key to responding quickly to software needs and establishing for itself a strategic advantage. In recent years there has been increased interest in object-oriented (OO) methods as a quicker, more comprehensive and viable alternative to traditional (data and process) methods. This paper presents the first phase of exploratory research into understanding the issues relating to making the decision to adopt OO, the actual process of OO adoption and evaluation of OO adoption within the Australian context.

Keywords

change management, exploratory study, information technology adoption, IS development methodologies, object-oriented approach

INTRODUCTION

The 1990s and beyond is perceived and accepted by most writers as an age of continuous change. These changes are especially evident in the areas of business management and information systems. As businesses rely more and more on computing technologies and information systems development to cope with changing requirements, it is acknowledged that the traditional approaches to system development are inadequate in coping with these changes (Taylor 1992; Berard 1993; Brown 1997).

In order to overcome limitations of existing traditional SDMs, IS practitioners are interested in adopting better and more effective SDMs. A major candidate as a new methodology is OO - the methodologies and techniques based on object orientation. OO has been in existence for nearly 30 years, but it is only with recent developments in programming languages and supporting hardware that it is currently seen as a promising innovative method. OO now provides a viable alternative to traditional methodologies (Kay 1992).

There is limited knowledge of the factors that influence the decision-making process when organisations consider OO adoption, limited knowledge of the ways in which the system development process needs to be adapted for OO development, and limited knowledge about the way in which OO methods can most effectively be introduced (Dick & Rouse 1994). Despite the general lack of knowledge, many organisations are seriously considering OO

adoption, are in the process of adopting OO, have adopted OO in some form, or are now evaluating the productivity enhancements of OO (Barnard 1998). Clearly, there is a need for research to address the lack of knowledge. Research findings would contribute to knowledge and would enable the development of guidelines for improved practice.

This paper outlines an OO research framework and provides an overview of the published literature. As such the paper provides a solid introduction to IS academics interested in studying OO related issues; the paper is also of interest to practitioners by providing an objective look at the issues. Specific research questions are presented which will be addressed in the next stage of this research project.

The paper starts with a discussion of OO's promise as the appropriate SDM for the new millenium. Then, the current state of OO in the US and in Australia are outlined. This is followed by the presentation of a research framework that could be used to investigate OO adoption. Finally, issues for further research are put forth.

OO: AN APPROPRIATE SDM FOR THE NEW MILLENNIUM

Advocates of OO assert that OO shows much promise as an appropriate SDM for the new millennium because it:

- is a 'complete' methodology;
- handles systems of arbitrary size and complexity;
- handles dynamic and turbulent environments;
- shifts the focus from one-off development solutions to integrated organisational solutions, and;
- enforces standards in system development.

Utilising a 'complete' methodology. OO is a methodology that provides a step-by-step process that provides segregation of phases to reduce unnecessary complexity (Taylor 1992). At the same time, there is ease of transition between the phases of analysis to design to implementation. Because of this, it is advocated as a complete system development methodology that concerns analysis and design and not just programming and implementation.

Handling systems of arbitrary size and complexity. OO is equipped to handle size and complexity of systems (Taylor 1992; Brown 1997). The OO data model is a superset of all the preceding generations of data models and hence, is able to cope with legacy systems. Also, its all-encompassing data model is able to represent data models of any arbitrary complexity and size. OO's encapsulation of related data and procedures is an effective tool for hiding or revealing complexity, allowing for freedom of expressing differing degrees of complexity in a business system.

Handling dynamic and turbulent environments. OO fares well on the flexibility criterion (Taylor 1992). If designed well, the high degree of modularity enforced in OO methodology allows for changes to be implemented in parts of the business without affecting other parts of the business. Reusability of business objects through its inheritance hierarchy decreases the time needed to develop business systems (Berard 1996).

Shifting from one-off solutions to integrated organisational solutions. OO combines the effectiveness of both process and data methodologies, together with its ease in integration with organisational structures via its object relationships, work to move system development to integrated organisational solutions. OO assists in viewing and modelling the organisation in a 'real-world' context – the modelling of people interacting with business objects to produce meaningful reports for effective decision-making (Brown 1997).

Standards in a methodology. Differing views are offered here. The optimists (Taylor 1992) suggest that OO methodologies are ready for prime time and are no longer a promise, but a

reality. After all, the OO programming component has been around for more than 30 years. On the other hand, there are numerous variants of the OO methodology. Pessimists and realists (Adhikari 1996) assert that standards in OO methodologies are as yet non-existent and that such standards are a pre-requisite for wide-spread adoption of OO.

CURRENT STATE OF OO

OO has found increased acceptance and use in a number of commercial software development organisations, consultancies, business organisations and universities (Reed 1995). Research centres have sprouted from this need to understand and properly manage this paradigm shift in system development. This section presents a brief description of the current state of OO by first discussing the adoption and use of OO in the US, followed by a brief overview of OO in the Australian context.

Adoption and use of OO

It is claimed that approximately 3% of commercial software development organisations in the US have adopted OO (Yourdon 1993). Using the adoption categories identified in Roger's diffusion of innovations (DOI) theory (1995), this puts US adopters at the innovator stage of adoption.

In evaluating current OO adoption and predicting potential OO adoption in the US, Fichman & Kemerer (1993) used a unified framework called the Software Engineering Process Adoption Grid (see Figure 1). The vertical axis reflects Roger's DOI (1995) view of organisational adoptability (relative advantage, compatibility, complexity, trialability and observability). The horizontal axis reflects economic factors affecting technology adoption (prior technology drag, irreversibility of investments, sponsorship and expectations). These axes combine to form four quadrants, implying distinctive adoption trajectories. Empirical testing of the grid on OO adoption was conducted in 1997 using case research methodology (Fichman & Kemerer 1997).

Organisational adoptability	Low	High
High	**Niche** – Adoption will start out fast among adopters who are relatively insensitive to standards issues or who have optimistic expectations about future levels of adoption. But adoption will plateau at a position short of dominance because of a failure to achieve critical mass.	**Dominant technology** – The technology will be rapidly adopted as a dominant process technology. It will face relatively low barriers to individual or community adoption
Low	**Experimental** – The technology will need to evolve before it is widely adopted by mainstream organisations as a dominant technology	**Slow mover** – The technology will diffuse steadily but slowly because of the difficulty individual or community adoption.

Community adoptability

Figure 1 - Software Engineering Process Technologies Adoption Grid
(Source: Adapted from Fichman & Kemerer 1993, p.10)

In the US, empirical testing of OO adoption has been largely confined to the studies conducted by Fichman & Kemerer following the construction of the model. Fichman and Kemerer (1993; 1997) found OO methodologies to be at the *experimental stage* of adoptionand hence, will need to evolve before they are widely adopted as a dominant technology by mainstream

organisations. OO was also predicted to have low acceptance in large in-house IS organisations.

State of OO Adoption in Australia

There are a few publications outlining the level of OO adoption in Australia. Reed (1995) reports that "[a] number of consultancies and development houses specializing in the use of OO techniques have arisen, and an Australian commercial object-oriented language (Ochre) has been produced". The general acceptance of OO approaches can be judged from the number of commercial (non-academic) OO conferences run every year and the steady increase in empirical studies of OO software practice between the years 1985 to 1995 (Reed 1995).

Dick & Rouse (1994) used Fichman & Kemerer's (1993) grid to study OO adoption in the Australian context. They conducted an exploratory study of four organisations in Sydney and Melbourne. According to Dick & Rouse's findings, OO in Australia has reached the *niche stage* – adoption will start out fast among adopters who are relatively insensitive to standards issues and/or those who have optimistic expectations about future levels of adoption.

RESEARCH FRAMEWORK FOR INVESTIGATING OO ADOPTION

The topic of OO adoption can be divided into three sub-topics corresponding to three phases in the overall OO adoption process (see Figure 2):

- The pre-adoption phase which includes issues affecting the decision to adopt OO. This phase ends when an organisation takes a decision about whether or not to adopt OO.
- The process of OO development when OO has been adopted as the appropriate method to be used in an IS project. This phase ends when the project is completed and software is implemented.
- The post-implementation phase which starts includes evaluation of the specific project developed with OO methods and the evaluation of OO adoption at large.

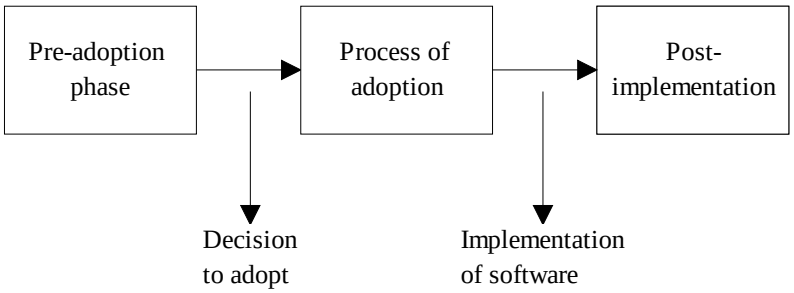

Figure 2 - OO Adoption Research Framework

Each of these three phases and the relevant literature will now be discussed in turn.

Pre-adoption considerations leading to OO adoption decision

This section discusses the issues that decision-makers would have to consider before making the final decision to adopt OO. First, the perceived benefits of adopting OO are discussed, followed by the perceived shortcomings of adopting OO. Then, strategies to address costs are addressed, followed by a final comment on making the decision to adopt or not to adopt OO.

BENEFITS OF OO APROACH

OO SDMs have received attention in information systems development due to the advantages over traditional approaches claimed by OO proponents (Wang 1996; Hantos & Joseph 1997). Numerous articles based on practical experience and empirical work have detailed the promises that drive organisations to jump on the OO bandwagon (Dick & Rouse 1994). The main benefits identified in the literature are as follows.

Increased reuse in system development. OO concepts of inheritance and polymorphism encourages the reuse of existing objects. Component-based development is performed at a fraction of the time and cost of traditional system development (Meyer 1987; de Champeaux & Faure 1992; Taylor 1992; Hantos & Joseph 1997).

Greater resilience to changes in the business. Low coupling and high cohesion within and between objects increase the maintainability of OO systems making OO systems more resilient to changes in the internal and external environments. OO systems are generally easier to modify and maintain than systems developed using the traditional SDMs (Henry & Humphrey 1992; Taylor 1992).

Faster system development. The notion of reusability and standard methodology notations, when implemented correctly, create highly reusable OO objects. Component-based system development infers a quicker and faster way to develop systems (Taft 1995). The promise of OO to eradicate problems of the mythical man-month of IS projects has contributed in a major way to its acceptance as "the" SDM of the millennium.

Better support for the creation of graphical user interfaces. The importance of GUIs in system development can no longer be ignored as most business systems today consider graphics as not only appropriate, but essential. Most OO-based techniques, tools and support tools are GUI-based to better support the creation of the GUI component of a system (Thompson 1989);

Production of higher quality systems. Because OO systems are largely constructed from existing, proven components, the methodology typically yields a system that is assembled from high quality, accurate and error-free objects. The end-result is a system that is of higher quality than those developed from scratch, as in traditional SDMs (de Champeaux & Faure 1992; Fichman & Kemerer 1992; Taylor 1992).

RISKS INHERENT IN OO ADOPTION

Although OO provides many exciting opportunities and benefits to a potential adopter, projects contemplating the use of OO carry some inherent costs (Taylor 1992; Hantos & Joseph 1997). One of the first questions that organisations ask when contemplating OO adoption revolves around whether similar risks are faced in both OO and traditional non-OO developments. The main costs and risks identified in the literature are as follows.

Technology investment. Most IS projects face critical decisions in the use of new technology (Hantos & Joseph 1997). OO projects are inherently based on new technology. Adequate technology is needed to support the activities of OO system development. Perceived risks include the incompatibility of OO with legacy system development and maintenance (Taylor 1992), suitability of methodology to suit the business philosophy (Adhikari 1996), vendor support and the maturity of OO SDMs (Fichman & Kemerer 1993).

Staff training and experience. "A sufficient number of experienced people are required among managers, architects, and engineers for the overall effectiveness of the [OO] development team" (Hantos & Joseph 1997, p.52). There are two main overriding themes in the literature of OO adoption with regards to the adoption costs related to the development team: paradigm shifts in mindset (Hantos & Joseph 1997) and the learning curve to arrive at a satisfactory level of OO practice (Aranow 1992; Attewell 1992). Unskilled development

teams may render the OO system to be a worse system than if traditional SDMs were used (Nerson 1992).

Management education and support. Management style and technique will need to be reviewed when adoption OO. OO introduces new ways of doing things that may not be compatible with existing managerial practices. Also, the introduction of a new methodology, such as OO, requires a champion sponsor. Managers must realise the commitment required of them to support OO adoption. Three main areas would concern business managers when adopting OO. Firstly, reward structures must be re-assessed to encourage code reuse (Taylor 1992). Lines of code would no longer suffice as measurement of programmer productivity. Secondly, management must be prepared for new employee positions and relationships to emerge, including a designated 'objects librarian' to index, manage, instigate, promote and facilitate re-use and a new object consumer-producer relationship (Classe 1995; Taylor 1992). Finally, these changes that occur within the development team and the organisation, as a whole, requires careful management (Taylor 1992).

Commitment to reusability. One of the ultimate goals of adopting OO SDMs is the increase in the reusability of software components (Meyer 1987; Taylor 1992; Fichman & Kemerer 1992; Fichman & Kemerer 1993; Brown 1997). Even so, adopting OO does not guarantee reusability. Reusability is not a direct result of using OO SDMs, it is more a result properly managing the OO system development process. The OO development team must have a commitment to reusability. This means that developers should design for future uses, not just for the problem at hand. The object consumer must trust the abilities of the object producer to produce highly reliable and generic reusable objects (Taylor 1992). Classe (1995) speaks of the importance of a tightly-knit team for object-oriented development and this is even more crucial in enhancing system development through reusability. Systems are developed through the assembly of a whole group of cooperating components. Components must rely on each other to do what they are supposed to do (Aranow 1992). True re-use requires significant discipline and organisational commitment (Aranow 1992).

Table 1 summarises the main risks associated with OO adoption and provides a number of mitigation strategies suggested by the OO adoption literature.

Perceived costs	**Risks**	**Mitigation strategies**
Technology investment	Incompatibility of technology	Pilot new technologies on a small scale before adopting them on large projects.
	Suitability of methodology	Use S-curve analysis to predict the future usage and popularity of a methodology. Also, OO standards-making organisation such as OMG's recommendations on specific methodologies could be a starting point in the selection process.
	Continuous vendor support	Popularity of the vendor, evaluation of the vendor's product and articulation of support terms.
	Maturity of OO SDMs	See "Suitability of methodology"
Staff training &experience	Paradigm shifts in mindset	Provide foundation-level training (OO concepts) up front before project-specific training (OO implementation language).
	Learning curve	Create an environment to learn incrementally and through customised mentoring programs.
Management education	Outmoded reward structures	Re-evaluate reward structures with a focus on reusability.
	New structure of employee relationships	For new staff, take personality trait into consideration in the employment process. For existing staff, staff education and reward structure may assist in the transitory phase. Also, there may exist a need to re-evaluate organisational structure, hierarchy and relationships.
	Change management	Negotiations and a close relationship with management through constant communication.

Commitment to reusability	Paradox: producer vs. consumer	See "New structure of employee relationships"
	Enforcing re-use	See "Outmoded reward structures"

Table 1 Perceived costs and mitigation strategies (Adapted from Taylor 1992; Hantos & Joseph 1997)

Making the decision to adopt OO will require in-depth analysis and evaluation of its benefits and costs (see Figure 3). Costs and risks are associated with any type of adoption but decision-makers have to be assured that these costs can be mitigated, or at the very least, be reduced to a satisfactory level, to reap the intended benefits of adoption.

The benefits of changing from traditional to OO methodologies must exceed the costs. Most importantly, there is a need for a collaborative decision to be made by both development and management teams to see if the benefits of such an investment can be realised within the time- and cost-frame of the system development project.

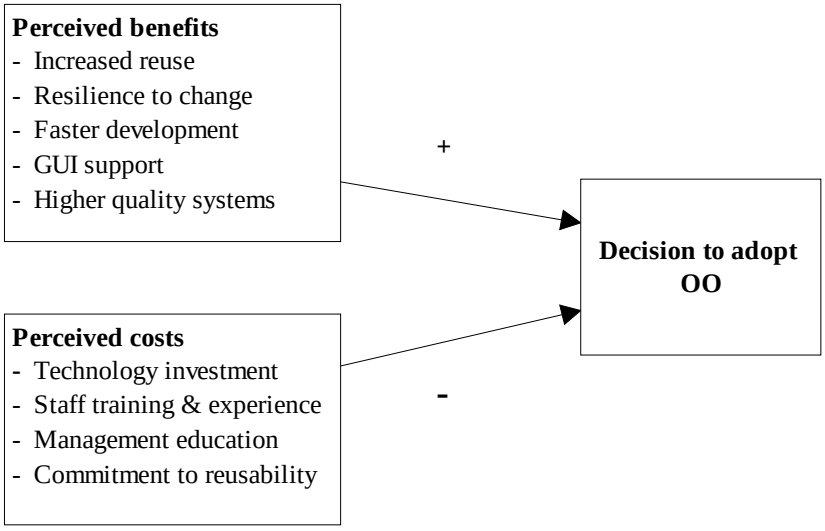

Figure 3 - Factors affecting OO adoption

Process of adoption of OO methods in an IS project

Once the decision has been made to adopt OO, the second phase in the adoption process starts: adoption of OO methods by the development team in an IS project. Few studies have focused on problems associated with adopting OO in a systems development project and hence, little is known about the strategies that Australian organisations use to address the process of OO adoption. The two main schools of thought concerning the process of OO adoption are discussed here. They are (Fichman & Kemerer 1992; Garceau, Jancura & Kneiss 1993) the revolutionary and the evolutionary approach; each will be outlined in turn.

Revolutionary Approach. Revolutionaries, such as Booch (1989) and Coad & Yourdon (1991) believe that 'object-orientation is a radical change that renders conventional [traditional] methodologies and ways of thinking about design obsolete' (Fichman & Kemerer 1992, p.22). Under the revolutionary approach, OO techniques take the place of traditional methodologies in a 'plunge' manner (Hardgrave 1997). The *revolutionaries* feel that the separation of data and processes as distinct entities are outdated. They feel that these

methodologies should be "thrown out" (Garceau, Jancura & Kneiss 1993; Fichman & Kemerer 1992; Yourdon 1993). Also known as 'complete rebuild' by Taylor (1992, p.293), it is simply to "pull out all the existing systems and rebuild them using OO SDMs." In this school of thought, the traditional methodologies should not be retained when adopting OO.

Evolutionary Approach. Synthesists or evolutionists, such as Wasserman, Pircher & Muller (1989) see object-orientation as 'simply an accumulation of sound software engineering [development] principles that adopters can graft onto their existing methodologies with relative ease' (Fichman & Kemerer 1992). Under the evolutionary approach, traditional methodologies are integrated with OO concepts to facilitate the adoption process (Hardgrave 1997). The *evolutionists* or *synthesists* (Yourdon 1991) attempt to integrate OO with the traditional methodologies.

Several authors (Aranow 1992; Fichman & Kemerer 1992; Taylor 1992) assert that the evolutionary approach has gained greater acceptance and has been more productive. Also known as 'graceful migration', Taylor (1992) assert that "[this is] the only realistic approach". There are several reasons for this:

- *Acceptance to change.* The introduction of OO in a gradual process increases the development team's acceptance to change in studying the new methodology.
- *Resources for retraining.* Off-the-job training is reduced as the development team trains on-the-job by integrating the traditional tools with OO tools.
- *Comparison of tools and techniques.* The on-the-job training facilitates increased understanding of the tools besides providing the team with a comparison of the effectiveness of the different tools in different situations.

Post-implementation evaluation of OO success

As soon as a software product developed using OO methods is implemented, organisations are interested in evaluating the success (or not) of the project and of OO adoption at large. Little is known about the actual benefits that have been realised by companies that have adopted OO. There is little evidence of actual productivity enhancements in OO projects. This could be due to the fact that the measurement of productivity enhancements in OO projects is significantly different from non-OO projects (Caspers 1994; Barnard 1998). OO attempts to increase productivity in all phases of software development (Caspers 1994). It also attempts to facilitate the creation of highly-reusable and high quality software in the development of future systems. Therefore, the timeline for measuring project success is markedly different than in traditional systems (Fichman & Kemerer 1993).

With systems developed using traditional methodologies, the measure of success is largely dependent on the end-users' perception of the systems. Some measures include user satisfaction (satisfying a need with an appropriate level of quality), ease of use, interest in the system and frequency of use (Barnard 1998). All of these measures are directed at only the newly developed system.

Although success of OO developed systems can be derived from the newly developed system, one aspect of OO is the reusability of its objects in future systems (Barnard 1998; Lorenz & Kidd 1994). The assumption is made that the objects developed in the current system have been tested rigorously to reduce the likelihood of errors and hence, are of high quality. These generic objects can then be utilised in future projects. The use of these generic and high quality objects is expected to increase the quality of future projects and, at the same time, reduce development time (Barnard 1998).

One way to measure project success is through the use of software metrics. However, standard metrics used with traditional methodologies are limited in their ability to describe

true OO analysis, design and code (Barnard 1998; Li, Cooley, Lewis & Henry 1990). Because OO success spans multiple projects, the issue of timeline and linkages between projects become a crucial issue. Measurements must be taken across projects, by determining the reusability of objects from one project to another.

RESEACH PROJECT

DATA COLLECTION

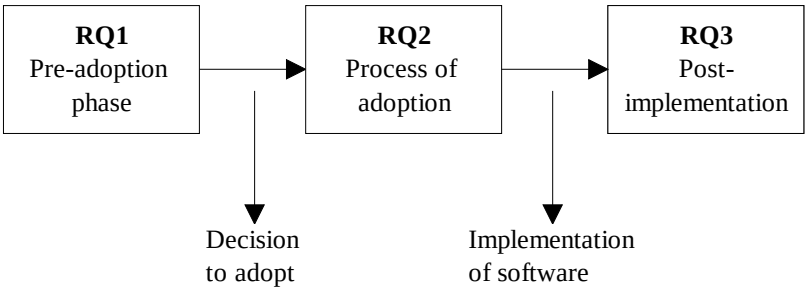

Figure 3 Research Framework for OO Adoption

An in-depth interview was conducted using the research framework summarised in Figure 3. Informants provided data on OO adoption in the most recent OO project that they were involved in. In-depth interviews were conducted with informants who have had:

- input into the decision to adopt OO;
- experience in assisting the organisation in making the decision to adopt OO.
- input and/or experience in assisting the transition process from traditional to OO methods;
- input and/or experience in the management of OO projects, and/or; and,
- input and/or experience in evaluating OO projects.

A large Australian government organisation was chosen as the initial case study for this research project. A project team consisting of 4 personnel and 2 consultants had just completed an OO project (referred to as "the project") at the time data collection was conducted. The informants were the IT Director, Project Manager and Analyst/Programmer. The use of multiple informants with differing job tasks and responsibilities provided different perspectives on the OO adoption process at QT.

The focus and results of each of the research questions is outlined in Figure 5-1.

RQ1: Why does a large Australian organisation choose to adopt OO?
This research question required an in-depth look at the factors affecting the decision to adopt OO, perceived risks OO adoption, and the possible mitigation strategies to reduce risks of adopting OO.

In the study, OO technologies were selected as a consequence of business and technology analysis. The project was regarded as an initial step towards achieving the organisational goals of encouraging reuse in system development, discouraging system development from scratch, and proactivity towards technological needs. The decision was affected by the evaluation of a number of factors. These factors can be classified into three categories: Very Important Factors, Important Factors, and Less Important Factors. From the in-depth interviews, the factors affecting the decision to adopt OO was ascertained (summarised in Table 1).

Table 2 Factors affecting the decision to adopt a technology

Level of Importance	Factors
Very Important	The technology chosen must: • comply with open, standards-based solutions; • not force QT to be locked-in to particular vendor(s); • be consistent with existing values and organisational goals/needs; • must produce visible results at the end of a project; • must show promise as a pervasive technology; and • allow for experimentation on a limited basis. Technologies that did not satisfy these criteria were not candidates for inclusion in the MPP.
Important	At the next level of importance were: • a large, mature base of existing adopters; • investments in the technology (including staff education, staff training and other financial investments); and, • the need for a champion sponsor to define, set standards, financially support and promote the adoption of the technology.
Less Important	Technical superiority over the its predecessors was considered unimportant in choosing a technology.

In assessing the perceived risks of adoption a technology for the project, the following were considered serious risks, in order of seriousness – from most serious to least serious:

1. The technology may require extensive training and education.
2. The technology may not enforce reusability.
3. The technology may not have stable implementation tools (programming languages).
4. The technology may not be compatible with existing systems.
5. The technology may not have a suitable methodology to support system development.
6. The technology may require high investments.
7. The technology may require organisational restructuring.

Contrary to the literature relating to remuneration of OO development staff (Federowitz & Villeneuve 1999; Classe 1995; Taylor 1992), the risk of a change in reward and remuneration of employees was "not applicable" to the project. The chosen organisation was a government organisation. As such, remuneration based on performance was not practised.

Mitigation strategies. Of the large number of mitigation strategies suggested by the literature (Federowitz & Villeneuve 1999; Heist & Allen 1999; Hantos & Joseph 1997; Fichman & Kemerer 1992), the organization used two main strategies to reduce risks of adopting OO:

- A small scale, pilot project to test out the adoption process and reveal the issues relating to OO adoption;
- The use of external consultants to aid in education and training of internal staff to sharpen their skills and reduce the impact of a paradigm shift in mindset.

Making the decision. A steering committee made the decision to go ahead with the project. At this stage, a go-ahead for OO technologies was given, but specific requirements and proprietary products were not selected yet. This steering committee consisted of members at various levels of the organisation and backgrounds (both business and technical personnel). A steering committee formalised the decision and provided a formal organisational structure for supporting the project.

This part of the research question identified the process by which a large Australian organisation would evaluate the benefits and risks of OO adoption, to arrive at a decision to adopt. At the organisation, the benefits sought after in OO technologies were:

- Greater resilience to changes in the business than traditional system development paradigms.
- Increased reuse in system development than in traditional system development paradigms.
- Higher quality systems than systems developed in traditional system development paradigms.
- Reduction of system development time than in traditional system development paradigms

In answering this research question, the issues that initial adopters should consider and the hurdles that might occur in the pre-adoption phase of OO adoption were outlined. Also, the *why of OO adoption* was revealed by the potential benefits that OO would bring to the adopting organisation.

RQ2: What is the process used by a large Australian organisation in adopting OO?
This research question explores the issues affecting the OO adoption process and the strategies used to address these problems. It also investigates the use of evolutionary and revolutionary adoption strategies and why one is chosen over the other.

The main issue encountered in the project was the difficulty of one project manager to manage the MPP. Two project managers were appointed: a project manager within the organisation (managerial) and a consultant project manager (technical). This provided the technical and business mix needed to achieve the project goals.

The second issue was the need for extensive education and training for staff working on the MPP. This issue was handled by including external consultants (who were knowledgeable with the technologies) to provide JIT-training and JIT-education. Contrary to literature associated with the process OO adoption (Joseph & Hantos 1997; Classe 1995), there was little resistance to change and few problems with shifts in system development mindsets. This was due to the personality traits of project personnel, an enthusiastic attitude and a small, well-managed project, and the possibilities of career progression from skills developed from the project.

The third issue concerned the selection of specific technologies for the middleware architecture. The consultants recommended technologies to be included. However, these technologies must be compliant with open, standards-based implementations. All of the technologies chosen in the project demonstrated this compliancy.

The fourth issue concerned the testing of the systems. Component-, interface- and system integration testing were performed. However, two out of three systems that were being tested with the middleware architecture were testing systems, not production systems. This reduced the risks of disruptions to existing systems at QT. This was a risk mitigation strategy that was used in the process of adoption.

The fifth issue was documentation. It was found that even though documentation standards exist in the organization, they were not adhered to in the development of systems. In the project, documentation was a problem, because the OO documentation style did not sit well with the documentation standards at QT. This issue was not resolved.

Adoption Style. A proactive evolutionary adoption style was used by QT in adopting OO. Decisions on the technology were made on a proactive basis. The technologies chosen grafted onto and/or improved the organisation's existing practices and designs. There are two components to this adoption style: "proactive" and "evolutionary".

A proactive style ensured that the organisation would always attempt to stay ahead of its competition. "Staying ahead" does not necessarily mean adopting superior technologies. At QT, "staying ahead" is achieved by understanding how OO can contribute to the organisation's goals.

An evolutionary style ensured that the path of least resistance be taken when choosing technologies. With evolutionary style, the impact on existing staff's skills and experience would not be revolutionary. An evolutionary mindset ensured that a change in staff's skills be perceived as an enhancement to his/her existing set of skills. Also, legacy systems will be perceived to be enhanced, not replaced, by new technologies.

RQ3: How does a large Australian organisation evaluate OO adoption?

This research question attempts to determine how "success" of OO adoption is defined, including the procedures to measure this "success", how these measures are defined and quantified.

The project had success measured *during* the adoption process and *at the end* of the project. During development, a standard project management software package was used to monitor and control the project. At the end of the project, a post-implementation review was performed to evaluate the success of the project.

The post-implementation review defined the parameters by which success is measured. Three major measures of success for the project were: the project's completion within its allocated budget, the project's completion within its timing schedule, and the effectiveness of the project output. Effectiveness in the project outcome was defined as the achievement of the objectives it had set out to achieve, the achievement of its implementation objectives, and the achievement of all of the key criteria laid down for a middleware architecture (security, scalability and reliability, reusability and extendibility, auditability, and performance).

No formal metrics (traditional or OO) was utilised when evaluating success of the project. This was possible and acceptable for the project because it was a proof-of-technology pilot. However, in other OO projects, this method of measurement may not be ideal. Project managers should look into OO software metrics, as suggested by Berard (1996) and Caspers (1994) as operational measures for OO project success.

ISSUES FOR FURTHER RESEARCH

Insights into the adoption process for organisations that have successfully or unsuccessfully adopted OO can provide practitioners with information about the important/crucial factors that could make OO a viable investment. Since the area of OO adoption and transition has not been well researched in Australia, there is still much scope for contributing to knowledge by focusing a research project on the adoption of OO by Australian organisations. In Australia awareness and acceptance of OO technology is becoming more widespread. The increasing number of articles/papers in both academic and practitioner journals indicates that it is timely to re-evaluate options and re-think OO adoption in light of these factors. The current research project is designed to enhance the understanding of OO adoption in Australia; it is exploratory in nature and is based on US theoretical and empirical research.

The current study provides a starting point for future research into technology adoption. This section provides suggestions and implications for future research into OO adoption.

Modifying the research parameters. The same research could be replicated in other government organisations and other industries. The research could be replicated in private sector organisations and compared with the findings of the current study. Also, cross-country research on OO adoption might provide a useful basis for comparing different adoption processes across national boundaries.

Implementing a multiple case studies design. A multiple case studies design could be implemented using two categories of adopters: adopters and non-adopters and even successful vs. non-successful adopters. This would increase the validity and reliability of the study on OO adoption. It would also enable tests for literal replication and theoretical replication.

Implementing a quantitative research design. This study confirmed the factors that influence adoption, the risks associated with adoption, the mitigation strategies used and the perceived benefits of OO. This study also revealed a number of additional factors and issues. These factors/issues can be operationalised in a quantitative study, using surveys, to increase external validity. This could lead to statistical generalisation of OO adoption in the Australian context.

CONCLUDING COMMENTS

System development is critical to the effective use of information technology but is often, poorly understood and managed (Fichman & Kemerer 1997). As interest in OO system development peaks in Australia, it is timely and crucial to address this shift from traditional to OO SDMs: the decision to adopt, the process of adoption and related issues of implementation and success. Research into this area of adoption will serve to bring forth the issues encountered in OO adoption and might help to ease the transition.

This paper has presented a research framework to investigate the adoption of OO methods and techniques and has outlined research questions which will be addressed in the next phase of this project. The output of this research is expected to add to the existing knowledge in the adoption literature. Although models of innovation adoption have been developed in the past, including the Diffusion of Innovations (DOI) model (Rogers 1995) and the Economics of Technology model (Arthur 1987; Farrell & Saloner 1987; Katz & Shapiro 1986), none of the models relate specifically to OO adoption in Australia. Therefore, contribution in terms of OO adoption within the Australian context is expected. Furthermore, because the research questions will be addressed using qualitative research methods, the output from this research will enhance and add to the depth of understanding of OO adoption and may provided bases for future quantitative work.

There is little research providing guidelines to practitioners in the OO adoption process. Although Fichman and Kemerer (1997) have identified lessons from early OO adopters in the US, generalisatiliby of such US-based findings to the Australian context has been questioned (Dick & Rouse 1994). One aim of the current research is to generate of a set of guidelines that could assist Australian organisations in the adoption of OO. The outcome of this research should facilitate more informed decision-making by organisations who are pondering the investment into OO and could even serve to ease the transition process from developing systems using the traditional methodologies to the utilisation of OO methodologies.

REFERENCES

Adhikari, R. 1996, 'Object-Oriented Development - Method To The Madness - Standard could make app building easier', *InformationWeek*, 576, pp. 1-2.

Aranow, E. 1992, 'Object Technology Means Object-Oriented Thinking', *Software Magazine*, pp. 41-8.

Arthur, E.B. 1988, 'Competing Technologies: An Overview', in *Technical Change and Economic Theory*, G.e.al.ed. Dosi, Printer Publishers, London, pp. 590-607.

Attewell, P. 1992, 'Technology Diffusion and Organizational Learning: The case of Business Computing', *Organization Science*, vol. 3, 1, pp. 1-19.

Barnard, J. 1998, 'A new reusability metric for object-oriented software', *Software Quality Journal*, vol. 7, 1, pp. 35-50.

Berard, E.V. 1993, *Essays on Object-Oriented Software Engineering*, Prentice-Hall, Englewood Cliffs, New Jersey.

Berard, E.V. 1996, *Metrics for Object-Oriented Software Engineering* [Online], Available: http://www.toa.com/pub/html/moose.html, [Accessed 99].

Booch, G. 1989, 'What Is and What Isn't Object-Oriented Design?', *Am. Progammer*, vol. 2, 7-8, pp. 14-21.

Brown, D. 1997, *An Introduction to Object-Oriented Analysis*, John Wiley & Sons, New York, US.

Caspers, J. 1994, *Object-Oriented Programming Analysis, Design and Implementation Methods*, Computer Technology Research Corp., South Carolina.

Classe, A. 1995, 'An object lesson. (importance of a tightly-knit team for object-oriented development)', *Computer Weekly*, pp. 34-5.

Coad, P. & Yourdon, E. 1991, *Object-Oriented Analysis*, 2nd edn, Yourdon Press, Englewood Cliffs, New Jersey, USA.

de Champeaux, D. & Faure, P. 1992, 'A comparative study of object-oriented analysis methods', *Journal of Object-Oriented Programming*, vol. 5, 1, pp. 21-33.

Dick, M. & Rouse, A. 1994, 'Introducing object-oriented systems development processes: The experience of four Australian organisations', *The Information Systems Working Paper Series: Monash University*, pp. 1-8.

Farell, J. & Saloner, G. 1987, 'Competition, Compatibility, and Standards: The Economics of Horses, Penguins, and Lemmings.', in *Product Standardization and Competitive Strategy*, H.L.ed. Gabel, Elsevier Science Publishers, Amsterdam.

Fichman, R.G. & Kemerer, C.F. 1992, 'Object-Oriented and Conventional Analysis and Design Methodologies', *Computer*, pp. 22-39.

Fichman, R.G. & Kemerer, C.F. 1993, 'Adoption of Software Engineering Process Innovations: The Case of Object Orientation', *Sloan Management Review*, vol. 34, 2, pp. 7-22.

Fichman, R.G. & Kemerer, C.F. 1997, 'Object Technology and Reuse: Lessons from Early Adopters', *Computer*, pp. 47-59.

Garceau, L.R., Jancura, E.G. & Kneiss, J. 1993, 'Object-oriented analysis and design: A new approach to systems development', *Journal of Systems Management*, vol. 44, 1, pp. 25-32.

Hantos, P. & Joseph, S. 1997, Object Technology Adoption -- A Risk Management Perspective, Xerox Corporation, USA.

Hardgrave, B.C. 1997, 'Adopting object-oriented technology: evolution or revolution?', *The Journal of Systems and Software*, vol. 37, 1, pp. 19-25.

Henry, S.M. & Humphrey, M. 1993, 'O-O vs. procedural programming languages: Effectiveness in program maintenance', *Journal of Object-Oriented Programming*, vol. 6, 3, pp. 41-50.

Katz, M.L. & Shapiro, C. 1986, 'Technology adoption in the Presence of Network Externalities', *Journal of Political Economy*, pp. 822-41.

Kay, A. 1992, 'The natural history of objects', *Journal of Object-Oriented Programming*, Supplement 12-13.

Li, W., Cooley, C., Lewis, J.A. & Henry, S.M. 1990, 'Software Metrics and the Object-Oriented Paradigm'.

Lorenz, M. & Kidd 1994, *Object-Oriented Metrics*, Prentice Hall, Englewood Cliffs, New Jersey.

Meyer, B. 1987, 'Reusability: The Case for Object-Oriented Design', *IEEE Software*, pp. 50-64.

Nerson, J.M. 1992, 'Applying Object-Oriented Analysis and Design', *Communications of the ACM*, vol. 35, 9, pp. 63-74.

Reed, K, 1995, *Software Engineering in Australia...A Brief Overview* [Online], Available: http://insect.sd.monash.edu.au/sercc/karl.html, [Accessed 98].

Rogers, E.M. 1995, *Diffusion of Innovations*, 4th edn, Free Press, New York.

Taft, D.K. 1995, 'OOP, there it is: Remixing, reusing existing materials', *Computer Reseller News*, 647, p. 85.

Taylor, D.A. 1992, *Object-Oriented Information Systems: Planning and Implementation*, John-Wiley & Sons, New York, USA.

Thompson, T. 1989, 'The Next Step', *Byte*, pp. 265-69.

Wang, S. 1996, 'Toward Formalized Object-Oriented Management Information Systems Analysis', *Journal of Management Information Systems*, vol. 12, 4, pp. 117-41.

Wasserman, A.I., Pircher, P.A. & Muller, R.J. 1989, 'An Object-Oriented Structured Design Method for Code Generation', *Software Engineering Notes*, vol. 14, 1, pp. 32-55.

Yourdon, Inc. 1993, *Yourdon Systems Method: Model-Driven Systems Development*, Yourdon Press, Englewood Cliffs, NJ.

COPYRIGHT