Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1999 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1999

Reuse of Object and Component Technologies

Jane Fedorowicz
Bentley College

Denis Lee
Suffolk University

Reuse of Object and Component Technologies

Jane Fedorowicz
Bentley College
Jfedorowicz@bentley.edu

Denis Lee
Suffolk University
Dlee@acad.suffolk.edu

INTRODUCTION

Companies are increasingly requiring that new information systems development projects employ object-oriented (OO) approaches. The hottest new Web tools and languages have object capabilities built into them. Much of the movement toward the OO paradigm for systems development is based on claims of vendors and pioneers that adoption will lead to better and faster designs, more maintainable systems, and most audibly, reusable software.

However, to paraphrase Brown and Wallman (1998), object technology is neither necessary nor sufficient for reuse, or what they call component based software engineering. Reuse may be theoretically easier in an OO environment, but it is frequently cited in non-OO projects as well. However, proponents of OO have argued that (ease of) use is one of its main advantages. In a survey by the Cutter Information Group, 80% of the 120 responding companies cited reuse as a driving reason for adopting object technologies (Radding, 1998).

Definitions of reuse vary depending on the nature of the reused component. This study adopts the following definition: "Software reuse is the process of building or assembling software systems from predefined software components that are designed for reuse." (McClure, 1997). Under this definition, reusable components may be objects or program source code, reflecting a commonly held view of reuse. However, reuse programs may also incorporate software specifications, project plans, frameworks, or any other software project deliverable. While focusing on the synergies expected from the reuse of objects, this study also examines other aspects of OO and reuse as they independently inform today's software practices.

BACKGROUND

Several theoretical reuse frameworks have been proposed in the literature, including the very comprehensive effort in Kim and Stohr (1998). However, reuse has only recently begun to be investigated and defined in a field research setting. Several case studies have been published illustrating the benefits and costs of reuse programs and approaches. In one study, significant development time and monetary savings were chronicled at Schwab for their e-trade system, which also resulted in improved user response time, a key competitive advantage within their industry. Schwab's success was predicated on adopting a single object language, Java (Levin, 1998).

Reuse proponents frequently point to the Software Engineering Lab at NASA as a leading success story for reuse. Reuse rates from 75% to 96% are reported for their projects, with the caveat that cost to develop code for reuse is higher than the cost to develop code without reuse as a stated goal. However, total development costs and error rates have dropped dramatically with reuse (Basili and Caldiera, 1995). Other studies at companies like Travelers' PC Claims unit, IBM, MBA Technologies, and 20th Century Fox show similar patterns of benefits (Fichman and Kemerer, 1998; Radding, 1998; Ross et al., 1996; Rothenberger and Hershauer, 1999).

Many of these case studies are primarily illustrative in nature. However, a couple of them augment the case with an analysis of the activities noted at the site. Rothenberger and Hershauer test a software reuse measure based on lines of code in three kinds of component software at one site. They computed an overall reuse rate of 54%, with no reuse of surface structure components, 56.4% reuse in the middle structure, and 100% reuse of the deep structure.

In the most comprehensive field study of all, Fichman and Kemerer examine 15 projects at eight IBM sites. To their surprise, they found IBM's current reuse practice to be informal and ad hoc, rarely extending beyond project team confines. They also found that many organization-centric reuse activities (such as formal reuse programs and centralized libraries) had been disbanded due to low participation and overall staffing reductions. They propose a reuse model comprising four dimensions, wherein a suitable combination of organizational model, production model, incentives and control, and funding and cost management need to be synchronized to support a successful and systematic reuse strategy.

In a predecessor project to the current study (Fedorowicz and Villeneuve, 1999), we analyzed surveys from over 200 OO practitioners to ascertain their level of experience with OO tools and techniques, and also to assess their perceptions of the usefulness and benefits of OO.

Of particular interest, respondents reported expectations that objects they have worked with are or will be shareable and reusable. This opinion was stronger when OO was used in a greater number of Systems Development Life Cycle steps, when project size was larger, and when the respondent had greater OO experience. In other words, the more OO was used, the greater the expectation for resulting reuse practices. These findings bear out many of the relationships proposed by the Kim and Stohr software reuse framework, and illustrated by the case studies described earlier.

THE STUDY

In the current study, practitioners with extensive systems development experience were surveyed to measure more precisely their experience with OO and reuse tools and techniques, and relate perceptions of the benefits (and costs) that accrue due to the adoption of these methods. A major goal of this study is to identify what is being reused, whether reused objects are developed in-house or purchased from an outside clearinghouse, and under what circumstances they are reused. The study also covers many other patterns of OO adoption and reuse, which are not reported here.

A three page questionnaire was developed based in part on the earlier OO survey, several widely used MIS surveys on usefulness and user satisfaction (e.g., Delone and McLean, 1992), and from practitioner writings on reuse (e.g., McClure, 1997). It was pretested in an object-oriented design course taken by advanced M.S. in Computer Information Systems students. The final, slightly modified questionnaire was distributed in July, 1997 to about 1,000 systems developers. The majority of the mailing list was obtained from *PC Week*, and comprised experienced systems developers. A smaller number of respondents were from a mailing list of reuse specialists who had been participants in reuse conferences and trade associations. A total of 190 usable surveys were returned. This paper contains a descriptive analysis of the data.

PRELIMINARY RESULTS

Tables 1 and 2 give demographic data for the respondents and their companies. Respondents have considerable OO and reuse experience. They are seasoned IS practitioners. They are employed by a range of companies, spanning many industries as well as small and large organizations. Twelve percent report the existence of a formal Reuse Program at their company, 5 percent used to have one but it has been disbanded (as was the situation at IBM as noted by Fichman and Kemerer), and another 4 percent describe various other approaches to a formal program.

Respondents were asked to approximate the breadth of use of OO and reuse within their organization. Table 3 summarizes these responses. While 16 percent reported no OO efforts within the past year, 20 percent perceived that over half of their company's application development efforts were object-oriented. Only 8 percent perceived that their company did not reuse components in the past year, while 27 percent reported that over half or all applications contained reused components. However, efforts to design components expressly for reuse were not as prevalent. Twenty-one percent reported no components were developed for reuse, although this does not preclude components from being purchased from a component vendor or other source, as indicated in Table 4. Twelve percent reported that half to all of their in-house development of components were intended to be reused.

Source of components was correlated with percent of components developed for reuse in Table 4. Where significant, the results support the contention that those companies reporting that components are specifically designed for reuse produce a higher percentage of reusable components in-house, while those purchasing components do not design for reuse in-house as frequently.

Not all reuse activity revolves around development of new applications. In the sample, 51% of respondents reported that reusable components were intended for new applications. However, 19% also indicated that reuse was practiced when modifying existing applications.

The top five categories of reused items are indicated in Table 5. Overall, only four categories of reusable components were reported by more than half the respondents, including code (62%), data objects (57%), programming objects (55%) and subroutines (54%). It is clear that code segments of various kinds are perceived to be the most highly reused components, far exceeding development methodologies, tools and frameworks in organizational reuse programs.

CONCLUSIONS AND FUTURE RESEARCH

The tables in this paper give a brief introduction to the data collected in this study. Further analysis should help researchers and practitioners to better understand how and when to rely on object-oriented and reuse techniques, by establishing patterns of use and benefit observation. The study examines the issue of object reuse, which will enable the establishment of guidelines on best practices for systems development and maintenance, as well as projecting future trends in the marketing and sharing of objects and other reusable components. Clearinghouses for code, whether internal libraries or independent vendors, are just now appearing in the marketplace. Additional research and analysis will lead to prescriptions for critical success factors for providing such services, and suggest cultural changes that must accompany technical change in practice in order to truly achieve widespread reuse.

References Available on request from authors

An earlier version of this work was presented at the IRMA conference in May, 1998.

Respondent Demographics	Mean Value
OO experience	3 years
Reuse experience	3.6 years
Years in computer industry	15 years
Years in current position	5
Percent male	91%
Age	40

Table 1: Respondent Demographics

Company Data	Percent
Number of employees	
<100	24%
100-999	20%
1000-9,999	29%
>10,000	28%
Percent with Reuse Program	12%
Percent without Reuse Program	80%
Percent that used to have a program but don't need it anymore.	2%
Percent that used to have a program but it didn't work and was disbanded.	3%
Other Reuse Program response	4%

Table 2: Company Size and Experience with Reuse Programs

Reused Items	Percentage of Respondents
Code	62%
Data Objects	57%
Programming objects	55%
Subroutines	54%
Software Design	37%

Table 5: Top Five Categories of Reused Items at Respondents' Organizations

Organizational effort in the past year (choose one only)	Number of companies reporting that this percent of applications developed were object-oriented	Number of companies reporting that this percent of applications developed contained reusable components	Number of companies reporting that this percent of components were developed expressly for reuse
None	30 (16%)	16 (8%)	40 (21%)
< 25%	72 (38%)	65 (36%)	81 (43%)
25-50%	34 (19%)	37 (19%)	21 (11%)
50-75%	10 (5%)	21 (11%)	16 (8%)
75-99%	15 (8%)	13 (7%)	5 (3%)
All	13 (7%)	17 (9%)	2 (1%)
Don't know	16 (8%)	21 (11%)	24 (13%)

Table 3: Companies' Exposure to Objects and Reuse

Source	percent	Correlation with percent of components developed for reuse
Percent of reusable components developed in-house using special tools and environments	12%	-0.00
Percent of reusable components developed in-house using programming languages	49%	0.24***
Percent of reusable components purchased individually	9%	0.12
Percent obtained as shareware or in the purchase price of a development environment	12%	-0.19 **
Percent of components purchased in an object "library" or "package"	11%	-0.06
Percent obtained as shareware	1%	-0.05
Other sources (contracted, etc.)	4%	0.06
*** significant at .05*		**** significant at.01*

Table 4: Source of Reusable components