A case for adaptable applications software

by MARY WOODWARD

Associates Financial Services
Southbend, Indiana

and

PETER F. DiGIAMMARINO

American Management Systems, Inc.
Redwood City, California

ABSTRACT

Contemporary economic circumstances have sent many organizations that extend consumer credit scrambling to secure automated support for collection operations. The traditional alternatives, custom system development and packaged software, fall far short of being acceptable to most large credit-oriented organizations. The Computer Assisted Collection System (CACS) was originally developed as a custom system and has since been used as adaptable foundation software by many large organizations to secure essentially customized support at a fraction of the cost, time, and risk that would normally be required. This paper reports on the success of the use of adaptive software to fill this urgent need and lends credence to the theory that throughout the 1980s there will be a trend towards the use of adaptive software to meet business’ demands for low-risk, low-cost, fully functional and tailored software.
INTRODUCTION

Consumer credit privileges are among the most visible and popular services provided by financial institutions and stores. Recent government regulations, economic factors, and social trends have had a profound impact on the business of granting credit. Consequently, consumer credit operations are of paramount importance in many of today's business organizations.

A variety of support functions are required to establish and maintain a profitable consumer credit operation, including credit authorization, accounting, customer service, and collections. Changing economic conditions and restrictive legislation often strain an organization's capacity to provide effective and efficient credit services. The high payroll and record-keeping costs due to their labor-intensive nature also significantly affect productivity and profitability of credit operations. These factors have an especially severe impact on credit collections operations.

The Computer Assisted Collection System (CACS) is a software system that improves the productivity and effectiveness of credit collections through a form of office automation. CACS provides users with immediate, on-line access to pertinent account information to assist in executing collection tasks and in making decisions.

CACS was developed originally as a custom system by Wells Fargo Bank, N.A., in cooperation with American Management Systems, Inc. (AMS), a company that specializes in management consulting and computer systems development. The system first became operational in the Wells Fargo Credit Card Collection Department in the spring of 1980.

Wells Fargo, like many organizations, was severely affected by the recession of 1975. Collection operations were strained beyond capacity as the number of delinquent and overlimit customers requiring proper follow-up surged. Shortly after the recession had eased, Wells Fargo Bank resolved to develop an automated system for support of collections in order to lower processing costs, improve collection effectiveness, and accommodate surges in processing requirements in bad times, as well as to allow for aggressive growth in credit operations.

Efforts to develop an automated collection support system soon revealed that such a system would require careful human engineering, the application of design techniques not commonly found in contemporary automated support systems, and a technical architecture that, on the surface, appeared straightforward but, in reality, was quite complex. Several years and approximately one million dollars later, Wells Fargo was no closer to having an operational support system than at the outset.

A new, 18-month venture with AMS finally resulted in a comprehensive collection support system that, almost from the day of initial operation, started to pay back in terms of increased collection productivity (close to 100% increases in productivity were measured) and effectiveness (record low losses and delinquencies were experienced). Wells Fargo, had, after many hard years, succeeded.

THE CONCEPT OF ADAPTIVE SOFTWARE EMERGES

The systems developers thought that the underlying CACS design and technical architecture were sound enough and flexible enough to be used by other organizations to help them secure a collection system meeting their own collection requirements. CACS, while certainly not a traditional software package, could be used as baseline software around which a customized collection support system could be developed.

The merits of this logic were taken to be

1. Wells Fargo Bank's prior experience, and the experience of at least a half-dozen other organizations, indicated that it is difficult, expensive, time consuming, and risky to custom-develop a collection system.
2. There were no mainframe collection system packages available on the market.
3. Even if packages did emerge, the esoteric requirements found in large organizations, which have evolved over decades in the business, mean that the use of off-the-shelf software would require major business and operational concessions, which would be undesirable and traumatic.

For some set of organizations, CACS as foundation or adaptive software could, it was reasoned, be used to secure a significant head start towards developing a custom collection system. Based on available data, the development of a custom system for a large organization was estimated to require a budget of from $500,000 to $1.5 million for professional services over 18 to 36 elapsed months. With CACS, the same functionality could be accomplished for fees and services of from $100,000 to $500,000 in just 3 to 6 months.

On this reasoning, Wells Fargo and AMS set out to test their hypothesis in the market place. The target customers were large organizations in several market segments (banks, finance companies, service bureaus, retailers, etc.) who might be planning to provide automated support to collections.

THE INITIAL REACTION

Initial contact was made with several target corporations. As expected, many companies were making hasty plans to pro-
vide automated assistance to collections since the current recession was at hand and collection woes were mounting. While the need was well-established and immediate, the alternatives were not attractive. The options were to develop custom software or to conform to the terms offered by relatively inflexible, minicomputer-based collection systems that had recently emerged.

The concept of starting with CACS and building upon it was greeted initially by two types of response. Those organizations with large systems organizations tended to say that their company never bought packages because packages couldn't possibly meet their specific requirements, were not developed according to internal standards, and caused more trouble than they were worth. Organizations that would usually be inclined to consider off-the-shelf packaged software tended to focus on the fact that CACS was not actually a package at all because standard, well-developed user documentation, run books, acceptance test scenarios, and the like did not exist.

The response to these points was that CACS represented a different kind of software solution. CACS is not a packaged system. It is a set of software that can be installed and built upon to provide, in the end, a complete system tailored to the organization's unique requirements. The system, in the end, belongs to the organization; it is unique and is maintained by in-house resources.

The decision to acquire and use CACS finally depends on a comparative analysis of functionality, cost, schedule, and risk (see Table 1). Without disputing any given organization's ability to succeed in developing a collection system, we see a good deal of empirical evidence that it is difficult to do so. The number of abortive efforts speaks for itself. For an organization starting from scratch, therefore, the risk of failure is high. The time and cost associated with custom development are also high relative to that required for systems that start with adaptive software. Finally, since the software is adaptive and easy to work with, all desired functions, features, and requirements can be accommodated.

The final analysis of the adaptive solution includes user review of the system in operation and technical evaluation of the system's components. The spectacular success of the system at Wells Fargo Bank, and later at other sites as well, left users clamoring for immediate installation of the system.

### TABLE 1—Comparison of packaged software, custom software, and adaptive software

<table>
<thead>
<tr>
<th>Evaluation Criterion</th>
<th>Packaged Software</th>
<th>Custom Software</th>
<th>Adaptive Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time required to implement/install</td>
<td>Low</td>
<td>High</td>
<td>Moderate to low</td>
</tr>
<tr>
<td>Costs to install</td>
<td>Low</td>
<td>High</td>
<td>Moderate to low</td>
</tr>
<tr>
<td>Risk</td>
<td>Low</td>
<td>High</td>
<td>Moderate to low</td>
</tr>
<tr>
<td>Degree of difficulty</td>
<td>Low</td>
<td>High</td>
<td>Moderate to low</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Low</td>
<td>Variable</td>
<td>High</td>
</tr>
<tr>
<td>Closeness of fit to requirements</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Support</td>
<td>Rely on vendor</td>
<td>In-house</td>
<td>In-house</td>
</tr>
</tbody>
</table>

The technical elegance of the system's underlying design methods and technical architecture left systems personnel convinced that the software would be easy to work with and maintain. Though initial reactions to the concept of starting with CACS are often negative, the final decision is clearly in favor of the adaptive software because of its handling of the critical problems of development:

1. The need to reinvent the wheel is eliminated.
2. The organization's unique requirements can be met.
3. The risks are low.
4. The costs are a fraction of what they otherwise might be.
5. The time required is similarly reduced.

### CACS STATE PROCESSING

There are several distinguishing design components of CACS that have primarily accounted for its success as adaptive software. The most prominent of these is its capacity for accurately and completely modeling work situations that conform to the following:

1. The work function is to manage a set of items according to prescribed procedures (e.g., clients, prospects, accounts, patients, etc.).
2. Associated with each item are one or more events that can occur (phone call, letter, payment, check-up, etc.).
3. For each item, one particular event is scheduled to occur next (e.g., follow-up telephone call on the tenth).
4. This next scheduled event defines the item's state.
5. It is essential to track all events.
6. Each event yields one of a set of possible results (e.g., no answer, insufficient payment, broken promise, bad credit check).
7. Given an item in a state, when an event occurs that yields a particular result, the prescribed procedures determine the new state for that item (e.g., after three no answers send a letter and obtain a credit report).

The item's progression through a well-defined set of work situations in response to the results of scheduled events can be documented using a standard state-processing diagram or a conventional finite-state automata grammar. Examples of this abound both in the world of computational theory and in the world of collections. Figure 1 presents a simplified view of this from a collections viewpoint. Figure 2 presents an actual state diagram from one of the CACS installations. The system's states and processing rules are easily defined, set-up, used, and refined at a given installation (and between installations) using the CACS State Definition and Transition Tables, Parameter Tables, and well-structured application software. With this, the time required to understand, completely and unambiguously, and accurately document the business functions, and then to design in detail and implement automated support, is reduced to a mere fraction of that which would be required using a conventional approach.

The CACS State Processing facility, with the system's technical architecture (see Figure 3) and its other generic support
Figure 1—CACS state processor overview
Figure 2—CACS state processor, diagram from actual installation
Figure 3—CACS technical architecture
functions (including list processing, audit trail processing, and historical data tracking), define a technology capable of being used as the foundation for a set of adaptive software systems.

**IMPLICATIONS**

Based on the CACS experience, there should be many similar success stories developed around adaptive software in the future. The need for multiple organizations to independently develop large and complicated custom systems to address the same fundamental business need is behind us. The need to modify the business functions to fit the packaged software solution is similarly obviated. The trend will be for well-designed software systems to be used as the foundation upon which other self-sustaining software solutions will evolve.

CACS is one of the forerunners in the evolution of such systems. Its success with this strategy has been both swift and startling. In just 14 months from the time of its introduction into the marketplace, 10 organizations have selected it for their solution. Half of these are already realizing the benefits from an operational collection system. The remainder will be operational within just a few months. Over 25 other organizations are now giving serious consideration to the system. The success of CACS as adaptive software is already secured. The success of adaptive software as a trend for the 80's lies just around the corner.

**REFERENCES**