DOCUMENT RESUME

ED 410 930	IR 018 492
AUTHOR	Moldoff, David K.
TITLE	The Emerging Trends in Application Integration.
PUB DATE	1997-00-00
NOTE	12p.; In: Association of Small Computer Users in Education (ASCUE) Summer Conference Proceedings (30th, North Myrtle Beach, SC, June 7-12, 1997); see IR 018 473.
PUB TYPE	Reports - Evaluative (142) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Computer Mediated Communication; Computer Software Evaluation; *Computer System Design; Costs; Electronic Mail; *Futures (of Society); *Information Technology; *Technological Advancement; Trend Analysis
IDENTIFIERS	Standardization

ABSTRACT

The purpose of this paper is to project present trends in application development into the next decades. The revolution in information technology has reduced the cost of computer hardware as well as communications. Networks will consist of three kinds of specialized components: clients, servers, and processors. Standardization of component interfaces and message structures will replace the "one size fits all" Legacy system, reduce the need for human interaction to perform a task, and reduce the cost of modifications and enhancements. Inherited data structures in Legacy systems create obstacles to enhancement and new applications that require data not accommodated in the existing structure. What is required is an environment in which systems consisting of "Best-of-Breed" components, both hardware and software, can be assembled using off-the-shelf proprietary modules. To ensure the participation of all interested parties, including institutions and vendors, consortiums must assume part of the leadership in setting and achieving goals such as: atomizing application software, standardizing component interfaces, and developing rules for peer-to-peer applications messaging. Over time, a team of component products will replace their large, complex, centralized view of data processing being delivered in Legacy software. Those who seize the opportunity to advance messaging between peer components will be forging the next generation of administrative systems. (AEF)

The Emerging Trends in Application Integration

David K. Moldoff
Applied Business Technologies
4631 West Chester Pike
Newton Square, PA 19073
dmoldoff@email.abtcampus.com

Introduction

Information Technology is Advancing Constantly

Everyone sees and feels the symptoms of revolutionary changes as innovation after innovation is introduced, and administrators and managers have been forced to adapt the way they perform tasks. During the past 50 years, for example, we have gone from wires, diodes and punched cards to mainframes, mini computers, on-line terminals, networks, and PC's.

Every administrative process can be divided into events or steps and, each step can generate multiple messages that reflect the status of the transaction or event. Messages can have standard actions performed before, upon receipt, at completion or upon deletion. Until now, users had three more or less expensive options for acquiring systems to perform these steps:

Home grown systems;

Purchasing stand-alone (Component) systems;

Purchasing integrated (Legacy) systems.

Today, we are still adapting to E-Mail; networking, Internet, and Microsoft Windows as standards are evolving for desktop operating systems, Local Area Networks, and Corporate IntraNets. Web standards and the communication media (i.e. Internet) will shape the landscape for information system technology and office automation in the 21st century.

The purpose of this paper is to project present trends in application development into the next decades. How can application software developers reduce the cost of their software commensurate with the cost of computer and communication hardware? How can software be made more adaptable to unique user needs?

Forthcoming Innovations

The revolution in Information Technology has reduced the cost of computer hardware as well as communications. The growth of complexity and cost of application software have accompanied user expectations encouraged by the increasing power.

The inevitable next steps must include software cost reduction by means of:

Greater modularization, standardization, and interchangeability of hardware and software components and interfaces.

Development of standard message structures to request and transmit data packets between local and remote components.

Networks will consist of three kinds of specialized components: Clients, Servers, and Processors. To accomplish a typical unit of work, a Client might send a message to a Server requesting that the Server perform sub-tasks (such as store, replace, transmit, delete, etc. transaction data). The Servers might, in order to complete its sub-task, send a message to a Processor to perform an operation (such as sorting a file). When the requested action has been completed, the Server will respond with the results.

Standardization and Interchangeability

Standardization of component interfaces and message structures will have the following benefits:

Facilitate mixing and matching of "Best-of-Breed" components;

Reduce the need for human interaction to perform a task;

Reduce the cost of modifications and enhancements.

Acceptance of Client/Server technology requires the migration of 'code' into small interchangeable packets that support the standard message architecture pioneered by the Web and Internet. The rocket-like acceptance of JAVA, Sun Microsystem's new hybrid language, is supporting this component concept called applets.

Future systems will consist of 'Best-of-Breed' application software components, replacing the "one size fits all" Legacy system, built to serve a large family of functions set within an organization. Legacy systems had a primary benefit of cost sharing, at the price of slow response to the need for change. They also lack the function-rich features of component software developed under Windows, the Web, and the workflow model supported by messaging. New software components, designed with the 'Best-of-Breed' approach with messaging will dissolve the Legacy software system piece by piece by replicating the current functions and supporting a friendlier user interface making productivity more important than cost and centralization.

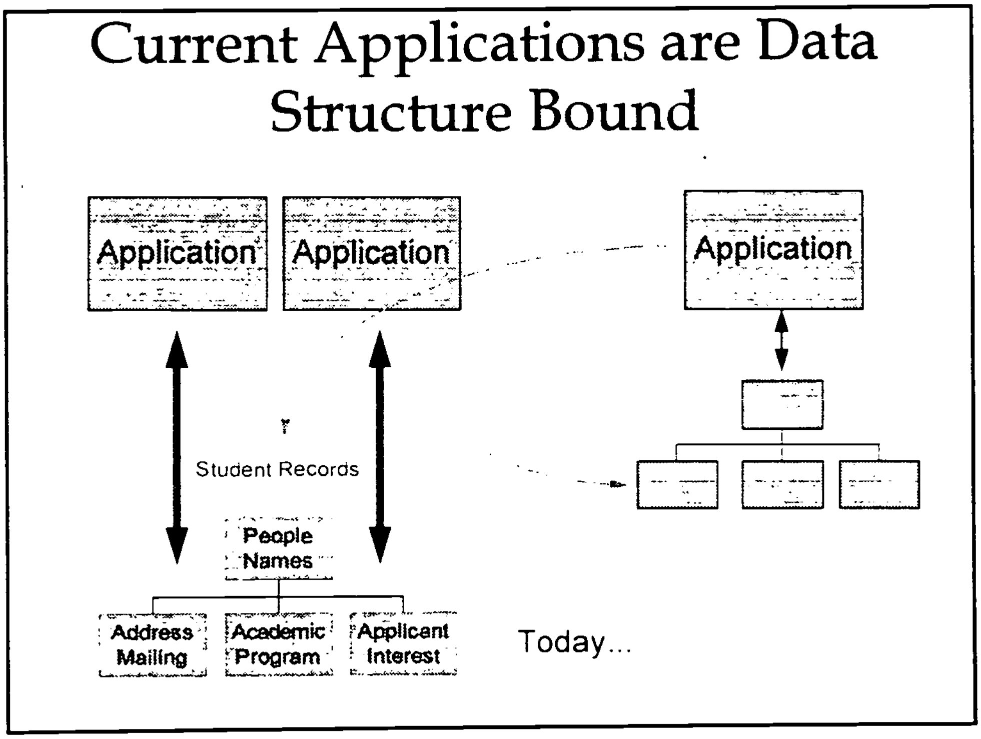
Cost, functionality, and flexibility justify migration to component-based systems. The next phase is for industry groups to define transaction formats to support Electronic Data Interchange (EDI) using electronic commerce as the justification to define the events that support messaging between and within Application Systems from point to point. The next step will be to design and construct communication components. Once communications have been established, there will be a surge of application development.

The Problem: Legacy Systems are Data Structure bound complex and Centralized.

In Legacy systems, most application modules are linked by shared data structures. Sharing data structures is not necessarily complex. Complexity derives from the number of applications, the number of data files and the interface functions needed to perform workflow procedures.

Inherited data structures create obstacles to enhancements and new applications that require data that are not accommodated in the existing structure. Two options are in order: (1) add new files, or (2) modify the existing files, with ripples throughout application designs. Both options add redundancy and increase the cost of operations as application components are implemented from different software developers.

A typical Legacy system may have hundreds of data files, which accommodate the need for many interfaces. Complexity is added when a shared data file is altered, impacting other application modules sharing access to the changed data file. The cost to accommodate changes in this model is high due to hard coded dependencies.



Current Applications are Data Structure Bound

The Solution

What is required is an environment in which systems consisting of 'Best-of-Breed' components, both hardware and software can be assembled using off-the-shelf proprietary modules. Interfaces between components must be integrated so that they appear seamless to users and managers.

To ensure the participation of all interested parties, including institutions and vendors, consortiums such as the Midwestern Higher Education Commission (MHEC) must assume part of the leadership in setting and achieving goals such as:

Atomizing Application Software,
Standardizing Component Interfaces, and
Developing Rules for Peer-to-Peer Application Messaging.

The criteria for the solution will be the ease with which it accommodates innovation and change, its inclusiveness, and its flexibility.

Application Software Atomization

Every application program can be partitioned into single-instruction atoms independent of the language or source of the program provided the installation includes translation tables. Each instruction message therefor needs, at the minimum, to identify:

addressee of the instruction,

the generic instruction,

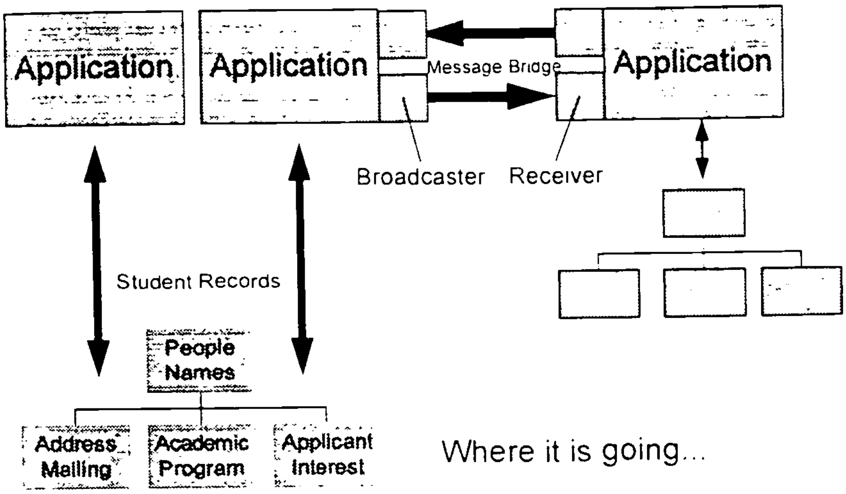
applicable identifiers,

operand source,

execution time, and

Disposition of results.

Applications with Messages Enabled



Who Will Set the Standards?

Legacy systems have implied standards set by the designers. Individual application components may even have internal standards, but they are all different.

For messaging to provide the bridging between independently developed applications, they must adhere to standards for:

Addressing and Protocol;

Message structure and language, and

Functional instructions.

When Thomas Edison invented the electric light bulb, he had to confront the need to distribute electric power. He chose a direct current option, but soon had to switch to alternating current because of lack of DC power transmission components. Much of Edison's investment in generating and distribution plant assets went down the tubes.

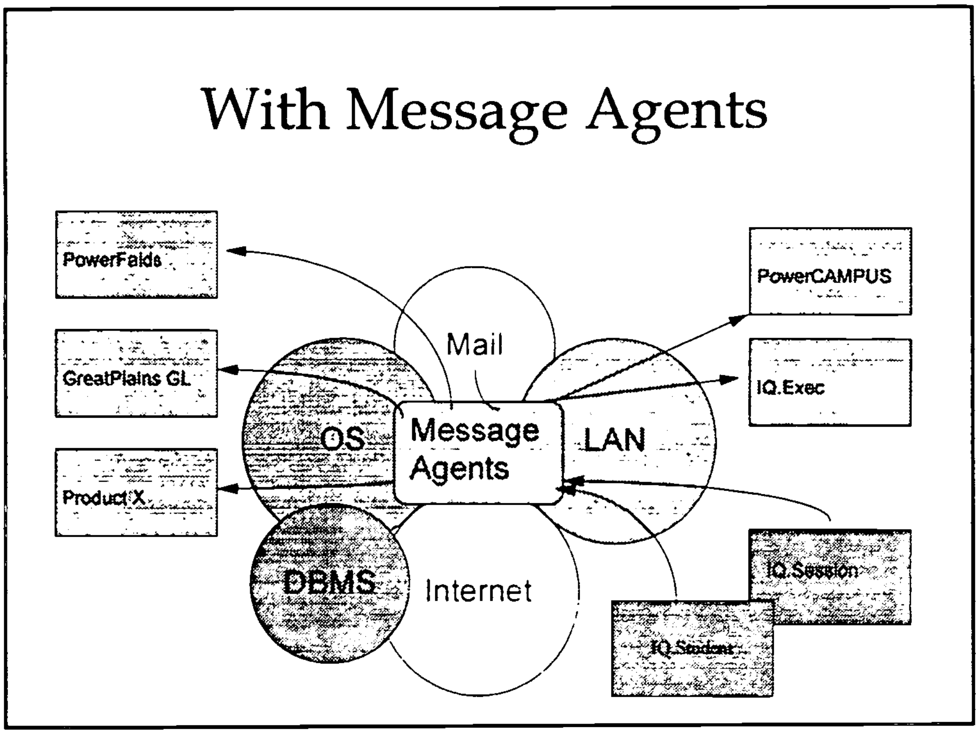
The challenge is different today. The information superhighway and its standards already exist. Transmission hardware and software are available. Lacking is the ability of application components to generate and interpret instructions and data comprehensible to disparate systems. The situation is analogous to a light bulbs made by various vendors that have different kinds of sockets and use different voltages and frequencies.

Industry Must Define Messaging Standards

Just as the industry is developing standards for reporting, it must develop standards for inter-application messaging to lower operational and acquisition costs. The longer standards are delayed. the greater will be the cost of implementation and integration of custom-built applications.

Both developers and users of application software canbe expected to agitate in favor of their own prototypes. Something like the FASB (Financial Accounting Standards Board) is needed to mediate and resolve the issues and define message standards.

This standard setting agency should have representatives of all of the interested parties lest it become captive of some dominating interest.


With Message Agents

Technology - Peer to Peer Messaging Solves:
The Bottleneck Problem and Creates a More Versatile Software Platform.

Each message is transmitted in a packet envelope which contains control data, such as source & destination identifiers, dates, transaction type & status, priority, selection criteria, security level, etc., and the messages, when saved, constitute an audit trail.

Network messages are deemed to be local, Intranet, or wide area. Messages can be described to be primary, secondary or dependent. And messages can be secured public or restricted. Messages need receivers (special programs in themselves) that can handle the methods, actions and behaviors designed into the message. This would allow application component products to be developed and deployed independent of their author and platform.

Instead of reading and writing into data files, applications transmit and receive electronic messages in standard event formats. Each Server listens in on the network for messages with its address, and captures those with the appropriate identifiers. (All other messages are ignored.) The Server places its messages in a queue and executes them in priority sequence.

Upon execution of the message, a message is returned conveying the results. The network medium is extremely fast, and the transmission delay is infinitesimal.

Messaging System Models

Electronic communications has undergone several revolutions since Morse code was invented. The development of the Web and Internet that is fueling radical changes in the computer industry are in the tradition of Bell's telephone, Marconi's wireless, Babbage's computer, and Sarnoff's TV. Key attributes that fuel the Web opportunity are:

The simplicity of the user interface;

The message driven model supporting the transmission of 'requests' and 'answers' from server to client;

The standard format of HTML documents;

The portability of the applications;

The ease of developing Web sites;

The links from Web site to Web site promoting cooperation and openness;

The graphics and text support making the products sizzle, and

The free form formats that combines structure with flexibility.

As the Web and Internet expand, the application model of computing is under pressure to adapt and facilitate electronic commerce via the open Web interface. Organizations worldwide are building their Web presence by altering business practices to include Web services. Microsoft and Netscape are battling for presence on the Web and control the tools that support the Web and integrate with the operating systems on the desktop and on the server.

Just as the Web was fueled by the above attributes, the platforms for business applications that run and administer businesses are under pressure to adopt the new model of information management. The evolution of computing has been altered by market acceptance of desktop innovations, cost savings in the use of purchased tools rather than in-house developments, and increases in access to information for a wider audience of users. These movements in the industry have been brought on by:

the migration from centralized computing to decentralized support for users performing the work;

decentralized functions on desktop computers that replace old ways of doing things, and

Integration of office tools on the desktop serving the productivity of users.

Application development lead-time and cost are major reasons for the market to encourage the trend toward the new model of computing. A Web message driven structure will reduce demands on centralized, complex Legacy systems.

Inter-Application Traffic

The same inducements for open systems between entities apply within administrative departments of each institution, and the market is shifting to components or object development based upon Microsoft's component object technology called OLE (Object Linking and Embedding). The concept seems complex, but is quite simple in reality. Build functions called 'clients' and 'servers' that mirror the application actions we perform as users. A 'client' requests data or services from a range of 'server' functions. These requests are enveloped in 'messages' that are transported between 'clients' and 'servers'. Windows is already based upon this technology and so is the Web. Most PC application software today must offer the same functionality.

A user fills the blanks in a screen form and accepts by using the OK button. Data are validated and sent to the database where they are processed. If an error occurs, a 'message' is returned to the function and displayed for the user in a box window giving instructions on what to do next. If the data are OK and accepted, the transaction is accepted and other functions are triggered, if necessary, to update other database tables.

The user filling in the screen form is in a sense using an object to perform a function. Today, the function resides in large central systems and called from menus. These functions are developed with instructions in Fourth Generation toolsets likeOracle, SQL Windows, PowerBuilder, or Delphi or legacy code like Cobol or Basic. The functions execute as 'fat clients' doing all the work or execute on a multi-user platform.

When functions are exported to standalone forms in the client/server model, they can be called at any point within the framework of the desktop OS. This way, functions can be seamless with the OS and interact with E-mail and the other supporting Office tools. This is one of the main thrusts of client/server and it enables the developer to distribute the workload between server and client desktop.

Most existing application software does not follow the approach described above since their designs rest on a foundation of data structures and complex logic performing the functions centralized in code developed over many years on UNIX or mainframe systems.

Legacy systems have evolved into complex, expensive and cumbersome platforms. They can be large bottlenecks to adopting procedural improvements. If we break down the complexity of a Legacy system by managing the events rather than the database transaction, we can reduce the costs of development of new functions and their deployment and at the same time it will give users new flexibility, control and productivity managing their work responsibilities.

The Transition

The transition to open systems has barely begun with use of E-Mail and Windows. Atomizing application software into component objects or little black boxes that perform single functions distributed over a network and shared by many users is the opportunity of the future. This trend has started with the acceptance of Windows as the standard desktop OS. ABT expects a major shift in application deployment to result in component objects as Legacy systems continue to lose footings in peripheral functions they perform poorly. In the next five to ten years, many organizations will adopt this system orientation because of their exposure to Windows on the desktop. Over time, a team of component products will replace their large, complex, centralized view of data processing being delivered in Legacy software.

The Agenda for Software Developers

A new model for computing will evolve because

The industry will be freed of software design restraints of record and table structures called relational database models. Organizations are building and implementing data warehouses to foster the integrated view of data while separating the operational layer from the data model. This will reduce the likelihood that one vendor or one product will support all operational functions.

Users will demand greater flexibility, simplicity, and satisfaction of their needs and expectations.

Institutions will demand faster response to their needs for changes at lower cost.

Data warehousing, messaging, and software atomization will facilitate modification and enhancement of application software.

Object oriented development, already used by Borland and Microsoft; will free code intensive applications from structural paralysis.

Messaging via the Web and Internet will liberate users from the constraints of geography and system structure. For example, student registration represents a sale to a student. The event consists of a course request, debiting the course offering, crediting the course-section list, and retaining the transaction as an audit trail. The process requires user based forms for on-screen interaction and message-based functions that send and receive data from the server holding the database structure. The Web has led the way in popularizing a new method to address how applications can be designed piece by piece offering a much more flexible solution for application development.

In the new scheme, non technical decision-making managers and administrators become users with the same ability to submit inquiries and receive pre-programmed responses without encountering delays inherent in acquiring formal reports through the chain of command.

The new environment encourages competition on the basis of quality and responsiveness. Institutions have the option to assemble networks of 'Best-of-Breed' components since no vendor (not even IBM or ATT) has all the resources to develop all the functions needed in the real world to satisfy every user in every organization. Expectations will continue to march forward and change and is the only action that can be predicted.

File/record imports and exports from function to function today support product integration. For example, when a prospect changes status to an applicant, a record is moved to a new structure. Or, when Cash Receipts are entered, the student's balance must be posted. The terms batch or posting is often used to describe a computer process that ties two or more functional areas together. This type of process adds extra steps to procedures and complexity. Moving to a dynamic message process to update across components, the real time updates would eliminate the need for these steps.

Criteria for Success

The union of high-speed data packet technology has opened the window of opportunity for system integration by means of messaging between applications on the desktop and server. This has made the linking between applications a common expectation never dreamed of a few years back. Those who seize the opportunity to advance messaging between peer components will be forging the next generation of administrative systems.

Those who bring out the `Best-of-Breed' solutions will win the race. These will be selected by astute educational institutions to assemble the effective networks of application software. As component products are built and employed, users will be able to employ suites of components to satisfy their functional needs and supplement many functions lacking or limited by design of older Legacy systems.

`Best-of-Breed' components are defined as those that are user-friendly, that are easy to implement, operate, support, and maintain; that have the functionality demanded by the users; and that perform effectively in networks with applications built by competitive vendors.

As organizations begin to adapt and work with EDI (Electronic Data Interchange) based applications; there will be a pull to adopt standard operating software supporting a message driven model independent of the database. This will provide a whole new level of computing. It will eliminate the application barriers present in large complex databases by breaking down the complex work process. Providing in essence, a component that does the work in a black box and delivering service or a solution in the message sent to a user. This may be in the form of a report, message box on the window or an e-mail message reminding a user to do something.

These developments will promote a new approach for administrative systems for colleges and universities and will force market consolidation. Vendors who continue to develop and support integrated administrative systems will need to cope with growing expectations of their installed base of clients and attempting to serve a continuously changing prospect pool who aspire to adopt new technologies rather than established products based on obsolete technology.

Many organizations have a great deal invested in present Legacy systems designed to maintain the organizations' data processing functions. The infrastructure of the Legacy systems is complex and expensive. Moving from Legacy systems to networks of application Components requires a significant investment in money and energy. Risk and stress are the price of advancing from one level of technology to the next; so many institutions are loath to replace their Legacy systems.

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

□ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket")..