



Integrating Information Systems

The Piece Parts Pitfall

Park City Group™
June 10, 1996

Contents

Introduction: System Theories	1
How Information Systems Get Selected.....	2
<i>In the Beginning</i>	2
<i>What We Inherited</i>	3
Patching the Pieces	5
<i>Getting It All Together: Integration and Implementation</i>	5
<i>Keeping It Going: Managing Multiple Systems</i>	7
On the Receiving End.....	9
<i>What the End User Wants</i>	9
<i>The Education Effort</i>	10
Pressures on the CIO	12
Abandoning the Piece Parts Trap	13
<i>The ActionManager Solution</i>	13

Introduction: System Theories

Software is designed to solve problems. When choosing software to manage business operations, companies typically assume that the “better” the application—based on criteria such as incorporation of the latest technological gimmicks, a laundry list of fancy functions, glowing reviews in trade publications, or a myriad of selection committee biases—the more effective the solution. And yet, best-of-breed applications continually fail to live up to organizations’ expectations. Why?

It’s simple. We have misunderstood the need we are trying to fill. Acquiring the “best” application—for example, the top-of-the-line E-mail package—isn’t the solution, because electronic communication isn’t the problem. The problem is *managing all of the company’s business operations in such a way that consistent execution of best practices, policies, and procedures across many business locations is ensured*. In other words, the organization’s technology platform must somehow accommodate all of the tasks and processes associated with the running of the business. It must provide for an integrated solution.

Purchasing the best E-mail package from one vendor, the best labor scheduling application from another, creates an information system where the whole is less than the sum of its parts. Although this statement seems counterintuitive, it is supported by classic system theory: You cannot optimize a system by optimizing its subsystems. A system must be considered as a whole rather than as individual pieces.

For example, you don’t buy a car by purchasing the most powerful engine from Ford and the latest electrical system from Chrysler—you would never get the pieces to work together. And yet, that is the technology situation in organizations today. A patchwork of point solutions has created a situation where the primary problem is getting everything to work together rather than providing the technological structure needed to manage the business.

The legacy of “focusing on the transmission rather than the car” is threefold. As piece parts are added to the organization’s systems:

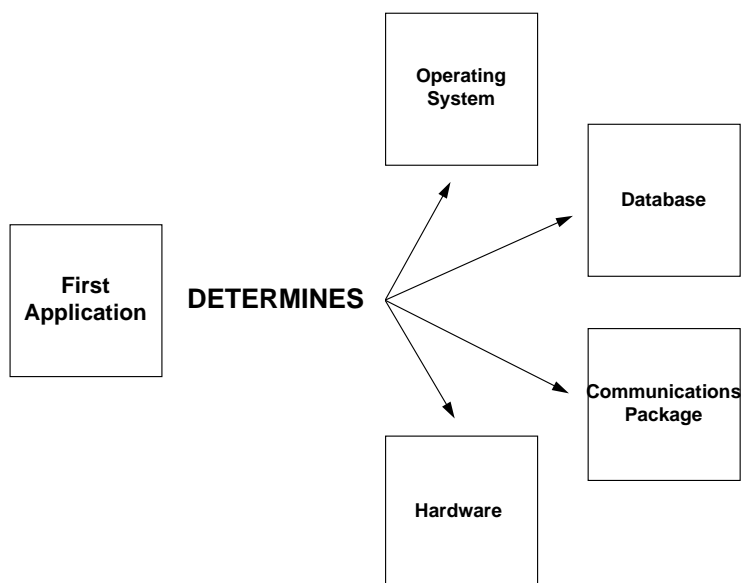
- Executives are constrained by existing software and hardware considerations when solving additional operational problems with technology
- I.S. professionals face an ever-increasing spiral of system management problems, during both installation of new applications and ongoing maintenance
- End users require more training per application and are slower to adopt new software

How Information Systems Get Selected

In the Beginning

There was a point at which no legacy systems existed in every company's history. Then the first package was selected and installed, probably using a process similar to the one that follows:

- Executives with a business management need identified a source for its solution: information technology
- Business requirements were defined and a Request for Proposal produced
- The application that most closely matched the company's business requirements was purchased
- Supporting systems were acquired based on the specifications of the purchased application



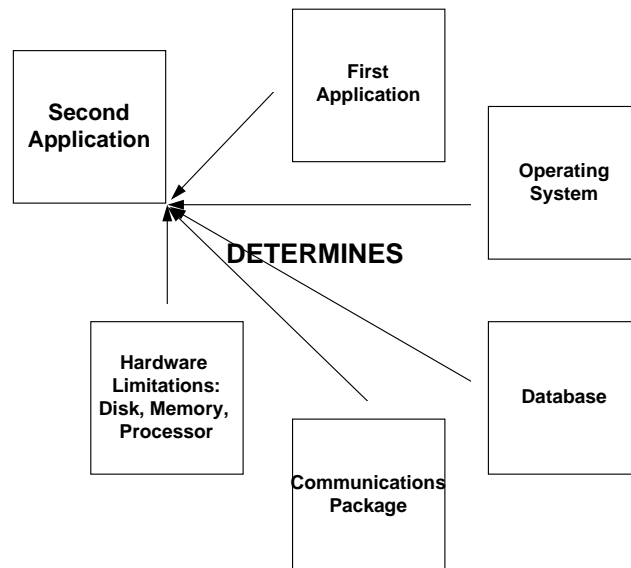
- A limited number of system components resulted in streamlined interfacing and testing phases
- The software was installed on equipment and shipped to geographically distributed locations
- All units were on the same platform, simplifying support of the application
- Trainers educated end users—a relatively straightforward process since there was only one package to learn

In this scenario, technology—and by extension, the I.S. department—is solely devoted to supporting the strategic needs of the business. There are no restrictions on what operational problems can be solved or the approaches taken to their solution. The I.S. professional is a hero, quickly and inexpensively putting executive vision into action.

What We Inherited

Of course, today's companies don't have the luxury of purchasing software in a vacuum. I.S. departments must cope with platform restrictions and a multitude of legacy systems. A typical selection process is outlined below:

- Executives with a business management need identify a source for its solution: information technology
- In addition to defining business requirements, platform requirements are outlined and incorporated into a Request for Proposal
- Packages that are incompatible with the company's existing platform are eliminated. Of the remaining applications, the one that most closely matches the company's business requirements is purchased



- Multiple system components complicate and prolong integration and testing phases
- Software must be installed on equipment at locations rather than by trained corporate personnel, creating rollout difficulties. Often installation responsibility is assigned to trainers, reducing end user training time

- Integration, testing, rollout, and support of the application is complicated by the fact that newer locations may be on different platforms than older locations
- Extensive training requirements include coverage of the new product, the differences between the new product and the old one, and any changes to old applications due to the integration effort

Although most companies today pursue a point solution I.S. strategy, it is clear that actually making a selection based solely on best-of-breed criteria is not an option. I.S. must focus on the tactical rather than the strategic. The overriding selection issue is that of identifying which vendor supports the organization's installed platform. And throughout the selection process, I.S. is forced into the naysayer's role, constantly asserting "It won't work with what we have."

The lengthy process of integrating and testing the new application with legacy systems, as well as the redirection of education resources from training to getting the software installed on-site further erodes the I.S. position. In this environment, the I.S. department is perceived as a stumbling block to achieving the goal, rather than the group that gets things done.

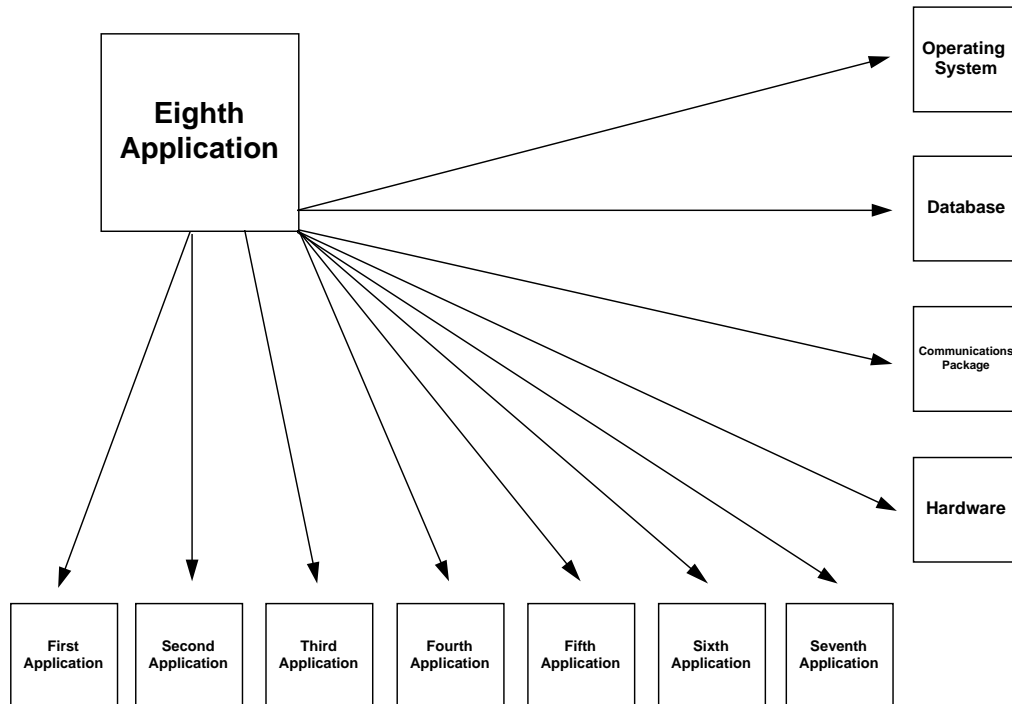
Patching the Pieces

If the selection of more and more point solutions results in an ever-narrowing number of options available to the executives responsible for wedding management vision and technology, the reverse is true of its impact on the I.S. department. More applications add exponentially to the time, effort, and cost expended on information systems. There are two primary areas of impact:

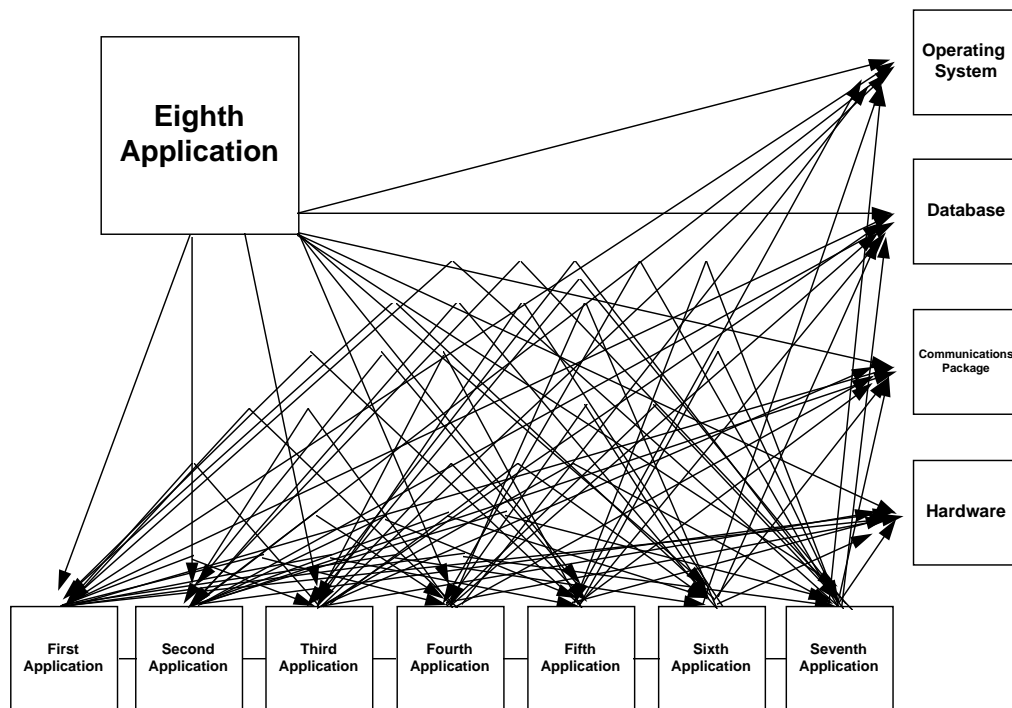
- The integration/implementation effort
- Ongoing maintenance and system management

Getting It All Together: Integration and Implementation

Why do piece parts create an integration nightmare for I.S.? One way to understand the problem is mathematical: The relationship between items being interfaced can be described in factorial mathematical terms. While managing any one item in the system is relatively straightforward, the situation becomes highly complex with the inclusion of three or four more.



As shown in this diagram, at first glance the addition of an eighth application to a company's information systems does not seem to be unmanageable. Of course, the new package must be interfaced and tested with the existing applications, operating system, database, communications, and hardware.



What is hidden in the first view of the problem is that changes must be made to the legacy applications in order to accommodate the new software. These modifications necessitate testing of each changed package with all other elements of the system, as shown above. As the system grows, the implementation of each new application is more complex and demanding than that of its predecessor.

In addition to being labor intensive, piece part integration poses unique difficulties. Applications from different vendors offer overlapping or inappropriate functionality without a mechanism for configuring the products to precisely meet specified needs.

New software requires newer versions of operating systems, databases, and communications. At the same time, existing software may not be compatible with these higher levels. The options are unattractive when it comes to managing this situation—upgrade the incompatible legacy systems, a solution which entails “from scratch” integration and implementation efforts for these additional applications, or run multiple versions of communications and databases, resulting in diminished hardware performance, increased memory consumption, and data duplication.

Of course, using many vendors makes it easy for each to say “It’s not my problem, it’s the other guy.” As I.S. pursues this frustrating merry-go-round, the business goes without needed functionality. Money—the return on investment anticipated from the software—and opportunity are lost. In the worst cases, the integration effort takes so long to untangle that when the application is finally rolled out to locations, it is already obsolete.

Keeping It Going: Managing Multiple Systems

The obsession with piece part systems has resulted in budgets that allocate 70 to 80 percent of total I.S. spending to maintenance. Multiple applications lead to multiple error logs, multiple audit logs, multiple security systems, nonstandard communications, and the inability to effectively juggle host job scheduling between competing programs.

A second strain on I.S. budgets is the ever-present threat of exceeding hardware resources. The “Murphy’s Law” of information systems in all probability reads “Each vendor will expand his product to consume or exceed available disk, processor, and memory requirements.” This pressure is multiplied, of course, by the number of different vendors installed.

But the greatest challenge posed by a piece parts patchwork is that of managing new version releases. As we have seen, new applications are sometimes the impetus for a round of software upgrades. The pressure from new vendors to go to higher levels of operating system, communications, and database software must be weighed against the instability of newer releases. Sometimes this pressure is premature. The company’s existing vendors may have different release schedules and currently offer no product that is compatible with the upgrades.

	1/1	1/1 5	2/1	2/1 5	3/1	3/1 5	4/1	4/1 5	5/1	5/1 5	6/1	6/1 5	7/1	7/1 5	8/1	8/1 5	9/1	9/1 5	10/ 1	10/ 15
1 - 1x	x																			
2 - 1x		x																		
3-1x			X																	
4-1x				x																
5-1x					x															
6-1x						x														
7-1x							x													
8-1x								x												
9-1x									x											
10-1x										x										
1-2x											x									
2-2x												x								
3-2x													x							
4-2x														x						
5-2x															x					
6-2x																x				
7-2x																	x			
8-2x																		x		
9-2x																			x	
10-2x																				x

Even without the introduction of new applications into the system, scheduling version updates poses problems. Consider a common scenario. You have ten products from ten separate vendors, each offering two releases during the year that change the product's underlying file structure. At what point in time do you start the interface work?

Depending on the vendors' release schedules you could have an application out-of-date roughly every other week, as illustrated above. When you choose your cut-off date for starting the project, at least half of the application versions interfaced will be six months old. It is not surprising that, given the cost and time involved, many companies choose to quit updating applications. The result? Existing technologies fail to keep pace with changes in the organization, new applications can't be added because of compatibility issues, and the entire system eventually becomes obsolete.

On the Receiving End

Corporate staff devotes enormous effort to the selection and maintenance of information systems that promise to improve business operations. In the end, however, system success is determined by the attitudes and actions of frontline operators. Simply put, a system that is not used has no value at all.

What the End User Wants

Up to this point, most companies have directed their attention only to the creation and improvement of end user education programs, neglecting the broader adoption factors that guarantee a system's acceptance and success. The common assumption that "If we show them how to use it, they will" is misguided. Education is only one element in the adoption equation.

Consider a typical end user—a manager in a convenience store. This manager has not received general computer training at school. She does not own a computer for home use. And unlike office workers, she has not been exposed previously to the point-and-click universe of GUIs. When she is first introduced to the store's computer, she must start at ground zero—the "On" switch.

The odds are that this store manager is intimidated by technology. She is cautious and uncertain when using applications, unwilling to experiment with the packages for fear that the machine will "blow up." Rather than perceiving the computer as helping her in her job performance, she feels learning to use and working on the computer takes her away from her true job of managing the store.

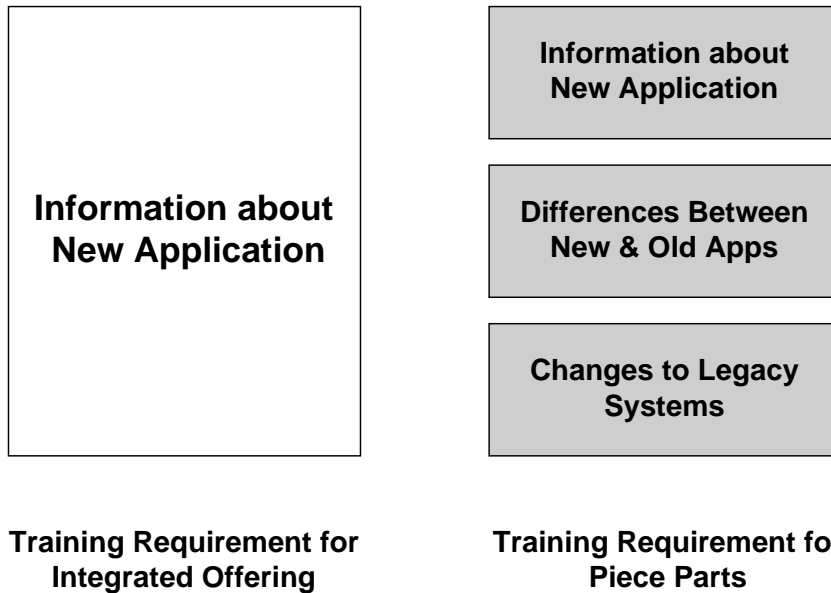
Given time and sufficient training, this manager may become proficient in the use of one application. Unfortunately, the time and resource required is prohibitively costly if the only education mechanism provided is roving trainers. In addition, between version releases and the introduction of new applications, her company constantly rolls out more products for her to learn. She is overwhelmed.

While training can and should help this manager, training resources are strained by the constant rollout cycle. In addition to training, another solution is needed—a system that incorporates "the way we do things" using language and processes that are already familiar to the store manager. Where computer commands must be learned (i.e., what happens when you hit the F1 key), they need to be consistent across every application in the system. In this way, as new applications are added, the learning process is one of building upon the manager's existing knowledge base rather than starting from scratch. She is more comfortable with each new application because it seems familiar.

Best-of-breed piece parts can't provide familiarity, which is necessary for rapid technology adoption. Point solutions vary in their tailorability to the company's methods of operations. Usually the end user must relearn how to do a task such as labor scheduling the way the software does it. Each package provides its own user interface, with a different look and feel and different navigation. The result is front line reluctance to accept new systems.

The Education Effort

A sense of familiarity is not sufficient for successful end user use of new technologies. Education programs do need to be developed and administered. Unfortunately, the piece parts model complicates this process, multiplying the information that must be covered by each program, while reducing time available for training on any one application.



In an integrated system featuring a common user interface and extensive tailoring capabilities, new product education must cover the business purpose of the application, when it is to be used, and its available features and functions. A piece parts system requires a more comprehensive agenda:

- Information about the new application—its purpose, times for its use, features and functions

- Differences between the operation of the new application and legacy systems—new methods of navigation, new function key assignments, etc.
- Changes—due to the integration effort—made in the operation of legacy systems

Piece parts force the user to learn more about each new application, while failing to reinforce learning that has taken place in the past. At the same time, two-thirds less time is available for training on the most important aspect of the new application—its business purpose. The piece parts education emphasis is on “how it works,” not “how it will assist you in running your business.”

Training is negatively affected by the piece parts rollout strategy as well, which relies upon installation by untrained location personnel. Because the new application must work on the location system before on-site education can begin, roving trainers become de facto installation troubleshooters. Every minute devoted to installation is time taken away from end user education.

An ideal solution to the training time squeeze would be to support on-site education with an automated method of self-instruction. Of course, in a piece parts environment, an automated instruction program is simply one more different and confusing application for the end user to learn.

Pressures on the CIO

The concrete ramifications of pursuing a piece parts strategy are clear—I.S. budgets volatile with unanticipated forced spending, significant I.S. resource devoted to a perpetual integration and rollout cycle, lengthy rollout periods and time consuming system management. In other words, I.S. projects that are consistently late, over staff projections, and over budget.

But there is a second, more subtle arena in which a piece parts strategy damages the CIO: the loss of customer satisfaction from his two primary constituencies, the executives who look to his work to support their visions and the front line operators who must use the end result. Executives perceive the CIO as the purveyor of tactical obstacles to what they want to accomplish rather than a technical strategist. End users experience a product that is complex, confusing, and often not relevant to their everyday needs.

As the initial promise of an I.S. solution to business operations problems fades, the pressures from both groups increases. Surely the next package will solve all our problems...won't it?

Abandoning the Piece Parts Trap

How do you solve the piece parts problem? With an integrated rule-based technology system that offers all the components needed to run your business operations. One approach is to build it yourself. Of course, the cost and staffing requirements of such an internal development project are astronomical. And in-depth experience in application definition and design is often lacking in a corporate I.S. staff.

It makes more sense to leverage the expertise of a software provider who offers such a solution, limiting corporate I.S. involvement to the installation process. Why build a system when the work has already been done?

The ActionManager Solution

Park City Group can provide you with the system that you would have built if you built it yourself. The ActionManager family of more than 30 applications offers modular solutions to virtually every operational need experienced by a geographically distributed organization:

- Workflow and communications
- Human resources management
- Labor forecasting and scheduling
- Information reporting
- Activity planning
- Cash management
- Inventory

In addition, a unique set of configurable action alert “boards” keep your front line staff focused on real-time conditions that require immediate response.

Products are quickly tailored to your specifications using your unique business rules. In other words, your company’s executive vision for how to manage the business is easily and exactly replicated in the technology. With business rules, end users experience a system that does things their way rather than vice versa. And a common user interface slashes training requirements.

ActionManager makes I.S. a hero with both executives and end users. And ActionManager makes I.S. professionals a success on their own turf. With one vendor to manage, the complexity of version releases and platform requirements is eliminated. There is one operating system requirement, one database requirements, one communication requirement, and one upgrade path. The guesswork is gone.

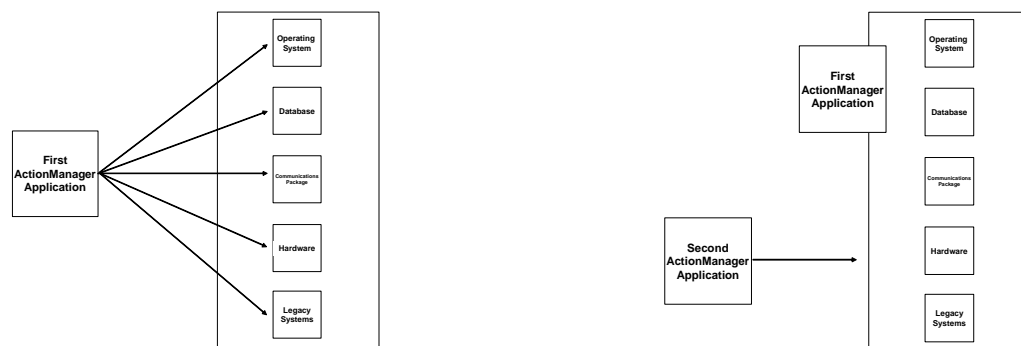
ActionManager's shared database ensures a single point of data entry. Because there is no redundant data, system resources are not squandered. More important, all information works together, providing a total, accurate picture of operational pulse points.

ActionManager's secret weapon is a shared technological foundation called ActionBase. The global functionality and "change once, changed everywhere" capabilities of this foundation greatly simplifies I.S. management of the ActionManager system.

ActionBase supports the following ActionManager features:

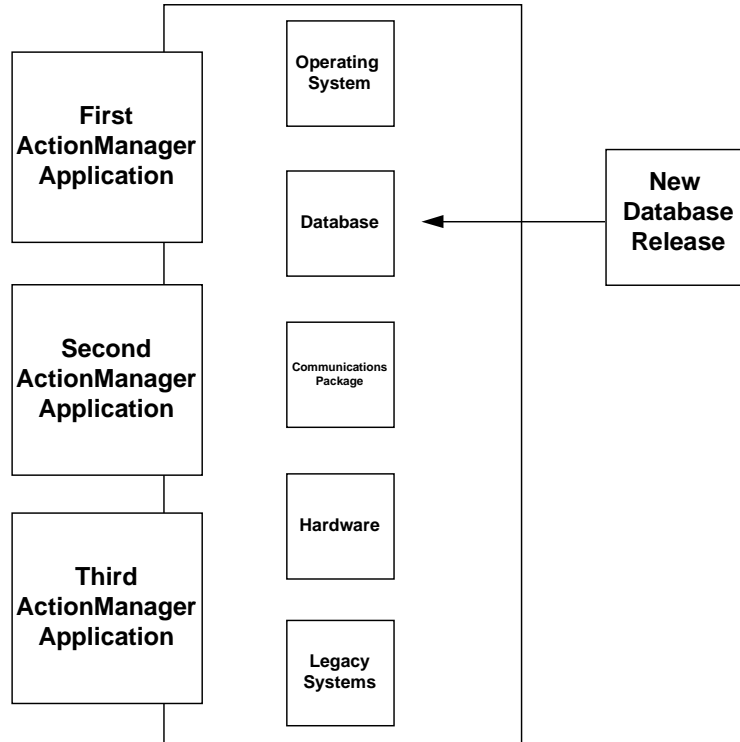
- Common application security
- Central error logging
- Central audit logging
- Job scheduling
- Information consolidation, summarization, and distribution
- "Specify once" interfaces with legacy systems
- A standard business organizational structure available to all applications
- Efficient use of system resources, shared memory for common functions

One way of understanding how ActionBase functions in the overall ActionManager system is to picture it as a "back plane."



When you acquire your first ActionManager application, you define your platform requirements and interface your legacy systems using ActionBase. Once the integration work is complete, any other ActionManager product can be dropped onto the back plane with virtually no additional effort. Because the integration effort is accomplished for the first application, each additional application requires incrementally less integration effort than its predecessor. Compare this to a piece parts scenario, where each successive application is factorially more difficult to install.

Likewise, maintenance of ActionManager applications is simplified by the back plane. Any shared specification need only be changed once—in ActionBase—to be immediately available to all ActionManager products.



ActionManager provides a flexible structure for rapidly responding to change in the business, whether at the application or foundation level. Designed and developed with one key question in mind—“How do you manage and deploy a distributed technology?”— ActionManager avoids the pitfalls of piece part solutions while elegantly solving your organization’s I.S. needs.