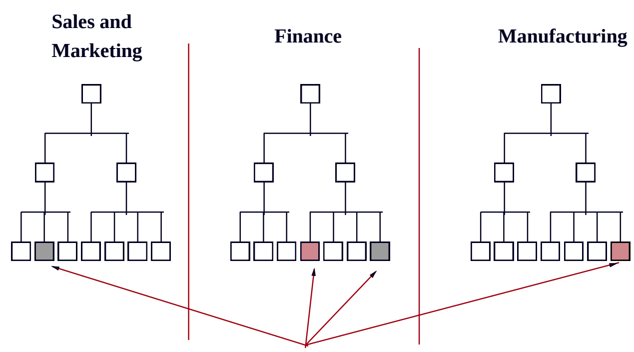
Process Integration Issues

By the Staff at Management Strategies

Most businesses organize their operations into multiple, concurrent operating processes. Often, these processes cut across traditional organizational boundaries such as departments or divisions. There are significant advantages to these cross-functional processes because there is a more natural relationship between the different work steps, especially in terms of relationships that impact or influence customer value. For example, the set of work steps necessary to complete a customer order, or handle a customer repair, naturally go together whether these steps lie in the Sales, Finance, or Manufacturing Departments. There is the danger, however, that the new process structure will simply constitute a different form of the old traditional “stovepipe” organizations.

Cross-Functional Processes



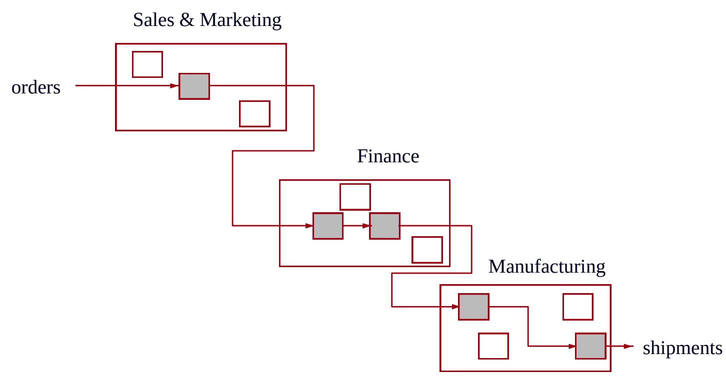
Cross-functional processes link subfunctions deeply embedded in different functional departments

These old “stovepipes” or “silos” developed because of the need for tight control over organizational resources, schedules, and budgets. Now, however, individuals and departments link together in process-chains that span these traditional organizations. If there are several such processes, each must have a process owner, and each must integrate smoothly and rationally with the others.

Business leaders need models that expose the linkages to view. Process maps are the accepted means for capturing details of the business environment and representing this information in graphical form.

Each of the functions in Figure 3-1 now has a place in a rudimentary process map that shows at least the input-output relationships among them.

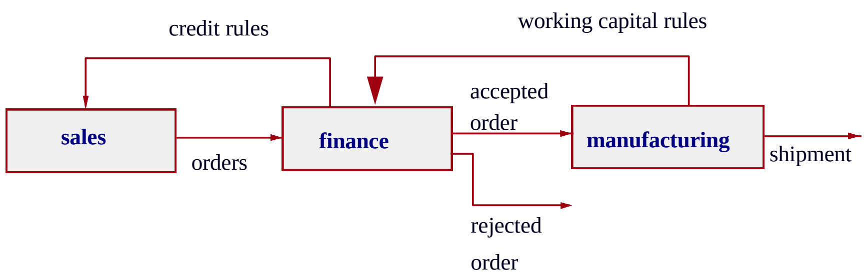
Models Partially Specify Business Processes...

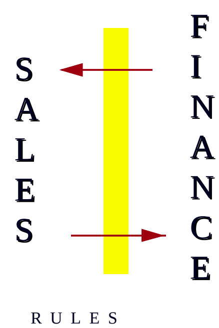


Existing process models expose business subfunctions and allow some control over "hidden" processes.

Process maps alone are not sufficient to fully explain how business processes work and how they integrate.

Models Partially Specify Business Processes... ...Rules Capture the Rest





sales-finance-rules: sales view of how finance works
finance-sales-rules: finance view of how sales works

configure complex product
product specialist

What Distinguishes ProcessA from ProcessB?

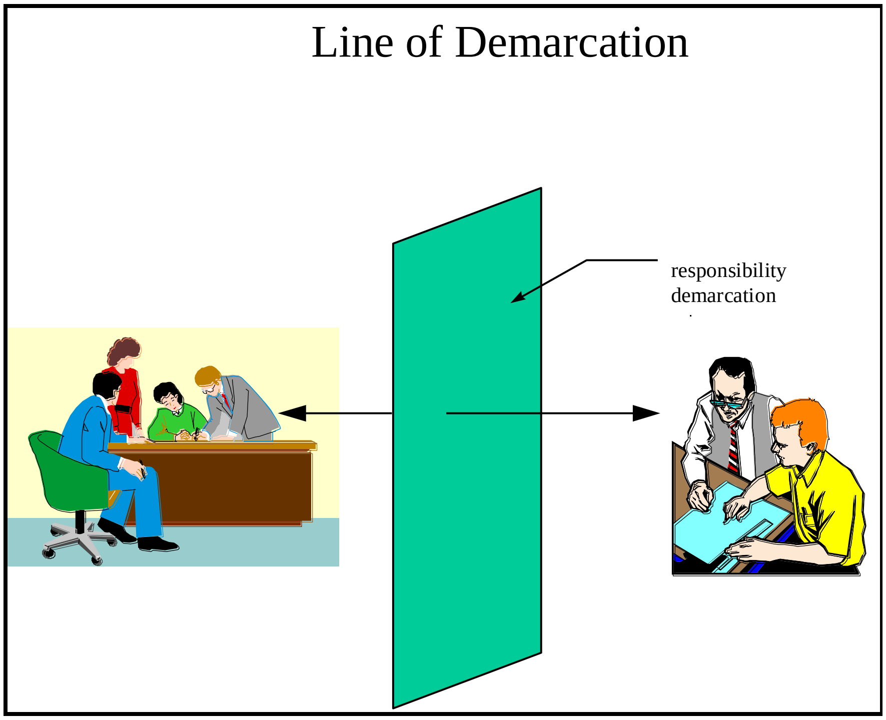
There are many ways that two or more processes can overlap or integrate with each other. The processes can share a common element, or have an input-output dependency, or can fall under the responsibility of a joint management team.

What makes a process element a member of ProcessA versus ProcessB?

The first principle is one of identity: the analyst must be able to distinguish whether a given process element falls within one process (say ProcessA) or another ProcessB. The business analyst or manager will intuitively classify a process step as belonging to FINANCE, or belonging to SALES. The ability to discern the differences, and membership in one process category or another, implies a boundary of some sort between PrcoessA and ProcessB resulting from a specific distinguishing characteristic. It is ability to draw this boundary and the distinguishing characteristics that defines an INTERFACE between ProcessA and ProcessB.

Usually the controlling management group establishes this INTERFACE, while other times it is a function of physical location. In yet other cases, it is a consequence some other sort of separation and associated security enforcement — for example, a border between countries, states, counties, or other agencies of governance.

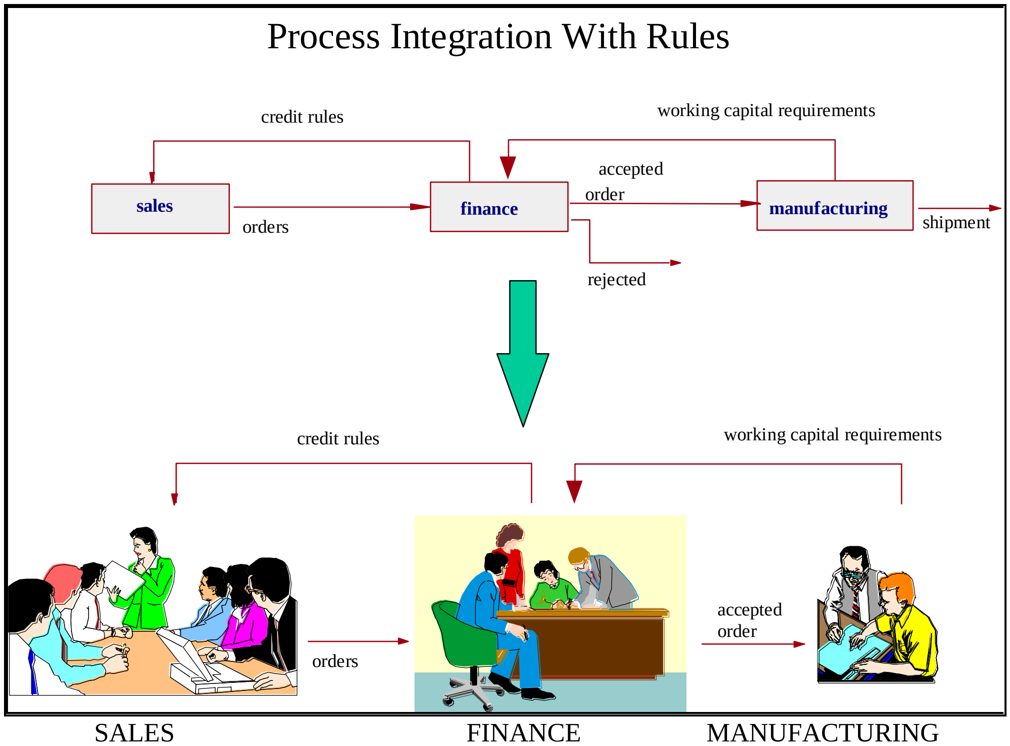
This INTERFACE also defines a separation of responsibility.



The two groups above are part of separate processes that interface directly to each other. Cross process integration implies mechanisms for handling complexities of relating activities across the responsibility boundary.

Crafting these mechanisms presents a challenge to most organization. To meet this challenge, many organizations resort to implementation-specific approaches that follow the business reengineering exhortation of "Focus on Outcomes!". Normally, implementation-specific views are generally quite helpful at various stages of business analysis, but for cross-process integration, there are many complexities that require a higher, "meta-level" discussion before such views can really bear fruit.

The nature of the unit operation that ProcessA represents can easily distinguish it from other unit operations that an analyst might otherwise lump together with it. This means that the kind of transformation ProcessA carries out to change its inputs to outputs, is sufficiently different from that which ProcessB accomplishes that most individuals (and certainly) managers will intuitively separate the two.



Much of the cross-process integration problem lies in the fact that the several involved parties must often act independently of each other in real time. If the SALES department had to coordinate directly with FINANCE on each and every order where financial issues came up, the costs and delays would be prohibitive. Yet without close and effective coordination, SALES will accept orders that FINANCE will wind up rejecting anyway, potentially tying up even more resources and leaving a string of dissatisfied customers who might never do business with the organization again.

Business rules management presents the most effective tool managers can utilize in the reengineered corporation. Rules provide a natural integrating mechanism. FINANCE promulgates rules that establish criteria by which the SALES department can accept orders, and offer credit terms in making bids and proposals. Similarly working capital requirements from manufacturing determines how FINANCE manages financial resources, takes out loans, and grants credit to customers.

That Which Links Two Processes

Integrating two processes means having to link them together somehow. The most direct linkage is when the output of ProcessA becomes the input of ProcessB. This is the only way for direct linkage to exist. Sometimes two processes are linked even though there are no direct input/output relationships between them, as is often the case in manufacturing assembly, for example in constructing a jet airliner, wing manufacturing will have no direct linkage to engine manufacturing. The two are independent (in terms of input/output relationships) until the final assembly units the two components in the airframe.

Other forms of linkage can depend on many different characteristics, each determining in one way or another how two processes interact with one another. These include:

Legal Organization

defines a clear boundary that separates two business entities although they may remain virtually indistinguishable from the viewpoint of final delivery: e.g., American Airlines and The SABRE Group.

Degree of Coordination

depends on the degree to which managers responsible for each process need to coordinate their production activities to assure acceptable final delivery.

Knowledge of plans: arms-length vs. team-play

The cross-process integration problem depends significantly on whether the individuals in the two separate process domains work at arms length, or as a team with detailed knowledge of each other's present as well as future plans.

Which 'side' of a financial transaction: customer *vs.* supplier.

Many times customers need directly visibility of a sequence of operations in the supplier shop, though for the most part this visibility need only extend to the primary process steps, and not the many secondary process steps, chains:

Workflow

The work flow defines an operating sequence, that is a series of work steps. The operating sequence is the management quantum. Generally speaking the operating sequence makes changes in state of a component or a part, or the relationships between two or more components.

Transient Responsibilities

Responsibility is one of the major integrative forces in weaving separate process work steps into a unified, seamless work flow.

Responsibility for specific items-of-work often must cross process boundaries. For example, when the steel service center sends out a quantity of metal for specialized finishing, cutting, or bending, the responsibility remains attached to the shipment. The outside processor assumes responsibility.

In essence, the steel service center says:

> "Here is my item-of-work. You, Mr. or Ms. Outside Processor, now have the responsibility for it, but I want my item-of-work back from you just as if I had cared for it myself. And ... here is what I want in terms of the specific cutting, bending, and finishing you are going to do it."

When the outside processor takes possession of the item-of-work, it assumes all the responsibilities for proper treatment thereof, including appropriate levels of insurance against loss, theft, or vandalism.

When the outside processor completes the order, and ships back the processed goods, the steel service center must validate that all is proper, and must have in place the proper quality control mechanisms as part of its receiving and inspection functions.

Dependent Responsibilities

Responsibilities that cross process boundaries are often highly dependent. For example, the steel service center salesperson negotiating on the phone with a prospective customer for delivery of a quantity of stainless steel sheet can't make a unilateral commitment. The sales person also needs to know the current production schedule, and the likelihood of producing the order by the promised delivery date. This information is generally well beyond the salesperson's immediate control, yet without these facts it may be extremely difficult to negotiate with the customer.

Though the sales process shares almost none of its process steps in common with production, nevertheless these two processes are joined at the hip in terms of the delivery-date commitments.

Under some circumstances (such as, diminished production capacity, staff illness) delivery is uncertain, and the company cannot make a firm delivery commitment. In these circumstances, the customer may want to know more about what is actually happening and may demand:

"Tell me what you're doing , and when you're doing it, so at least I can track your progress and determine what I am to do."

Effective integration with another process requires knowledge of the state of a process upon which a given work step depends. To achieve this information flow is critical.
Three kinds of information are necessary: status, control, forecasts

Status information includes knowledge to the state of involved elements. In the example of aircraft manufacture and assembly, the two groups working independently on wings and engineers need to understand the state of each other's work-in-progress. Generally there are numerous production rate decisions that each group must make.
If the engine contractor is proceeding slowly (because of a strike, or shortage of titanium alloy) there is no sense running the wing construction process at full tilt — that just ties up shift labor; and creates excess work-in-progress inventory. Similarly in the case of the
steel service center; the sales group and the production group need to understand the state of the production process and the state of incoming sales orders. If there is a queue building up for a specialized finishing machine or a cutter, production managers can adjust their work queues, add shifts, or defer orders that are less time sensitive.

Control information varies with the span of control for a group's organizational structure. Because of downsizing, reduction in middle management ranks, and other organizational paradigms, the span of control is increasing and organization are becoming flatter. This has enormous implications for the nature of information flow and changes in responsibilities. Unless adequate mechanisms exist for insuring control, the organization becomes susceptible to errors; and deviations in time; cost, quality increase. The next section on variance analysis makes clearer what some of these mechanisms should be.

Forecasts are a special form of coordination that involves not just the immediate time frame, and situations and sales orders under present consideration; but also future orders; indeed future markets. A primary example is that of product development involving integration of product engineering of market-driven features and processes for assuring dominance in specific market segments, and competitive viability. This includes specifying measures for future states outcomes, and determining how well forecasts of these measures track to actual results.

Functional Assignments

Functional assignments are the general mechanism for handling distribution of responsibilities in most organizations, but for cross-process integration significant care and attention may be necessary.

Often, even in traditional organizations, definition of work-group responsibilities may not match up well with the circumstances and events the group encounters in the ordinary course of doing business. Whose responsibility is it to root out the reason why a particular customer was irate: is it the Customer Service department that originally fielded the complaint, or Sales that sold the customer a substandard unit, or Manufacturing that built a unit that complies with what marketing wanted but didn't meet customer expectations.

Even when these responsibilities are clearer, or joint-responsibility teams exists, it is at times extremely difficult to achieve true empowerment of involved personnel so that they can catch problems and manage them to completed resolution. The next section illustrates a mechanism for determining how to achieve this by defining, analyzing, and exerting control over process variances.

Social-Technical Organization Design and Variance Analysis

Traditional organizational structures simply aren't suitable for the fast-paced competition of today and the faster-paced competition of tomorrow. We must redesign organizations so that they can specialize according to function or task-set, but integrate seamlessly for continuity of purpose and operation in serving customer needs. Process integration is most tractable when it is possible to develop good models of cause-and-effect, and penetrating to the heart of variations in performance and outcomes is the starting point.

Not all process integration factors are created equal. Some are inherently more sensitive to variations in quality, performance, and cost than others.

Variance Analysis is a way of unraveling the relative importance and impact of quality-affecting process interactions.

The underlying assumption for Variance Analysis that each business has many specialized decision points that affect quality, or at least have the potential for doing so. These Quality-affecting decision points QADP are often unclear because a myriad of details and other process interactions obscure their nature and function.

Variance analysis can also show where the most sensitive QADPs are located.

One example is the Revenue Stream Analysis methodology which represents one approach to process reengineering , as employed in companies like CPRail, Montreal, Canada. Under the Revenue Stream Analysis construct, analysts at CPRail modeled the firm's business processes in terms of inputs to (and outputs from) a series of Unit Operations. Each Unit Operation is itself analyzed for Variances, which are deviations from standards expressed in terms of Quality, Quantity, Time and Dollars. An important stage of Revenue Stream Analysis includes the assessment of each variance's impact on all other variances, in an effort to identify "key" variances, those whose "downstream" impact is significant.

Performing this type of analysis for any reasonably complex operation results (at least preliminary) in identification of a large number of Variances and associated Unit Operations. To ease the manual burden of tracking, managing and displaying these Variances, Management Strategies has developed its "Variance Handler" prototype, implemented in Microsoft's ACCESS database tool.

The Variance Handler is a tool that allows the user to generate models of operations and processes using the "Variance Analysis" concept. The Variance Handler allows users to:

- Create new models and load previous models
- Define, modify, and delete unit operations within a model
- Define, modify, and delete variances associated with each operation
- Define, modify, and delete linkages (or dependencies) between variances

The linkages among variances in different unit operations link them together, but only indirectly, by and the order in which they appear in the table of unit operations.

The Variance Handler allows the user to view and generate reports of information in several different formats. Users can generate the following reports:

- All unit operations in the model
- Variances associated with a particular unit operation
- Upstream and downstream variances associated with a given variance
- Control tables for selected key variances
- A Waterfall Diagram Table

In addition, unit operations, variances, variance linkages, and the Waterfall Diagram stored in the Access tables may be exported to a Lotus, an Excel spreadsheet , Word, or IDL format which can be imported by IDEF and several other modeling tools.

Control Table

The Variance Handler allows the user to create a control table for each variance designated as "key." Control tables are used to monitor and control key variances that affect process output. Control tables include information such as:

- Where the variance is observed
- Where the variance occurs
- Where the variance is controlled
- Who controls the variance
- Activities required to control the variance
- Information requirements related to control activities

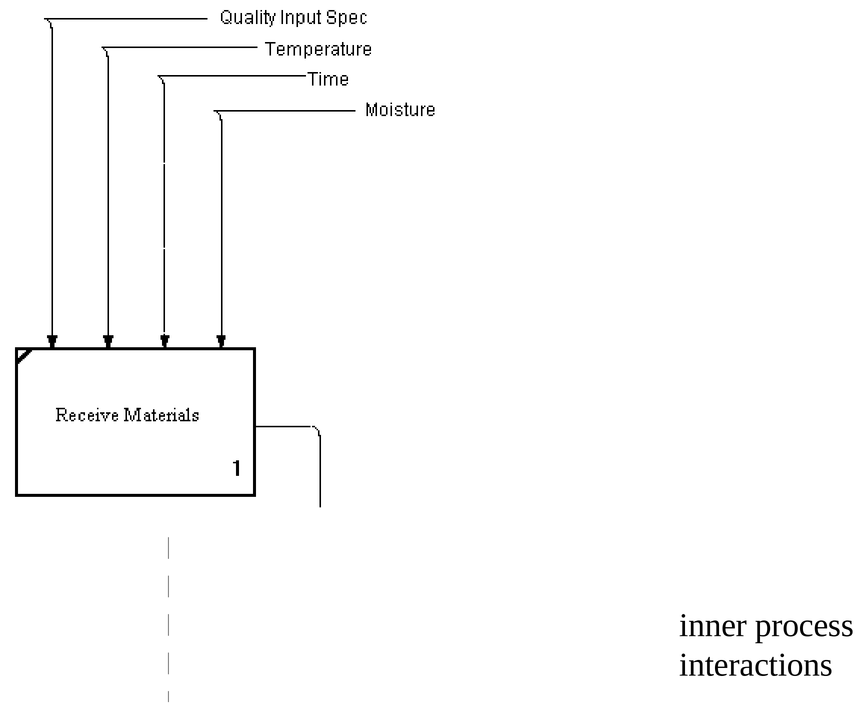
IDL Diagram And Waterfall Diagram

The IDL diagram that the Variance Handler generates provides a pictorial high level view of the process model. It only shows unit operations in the model and the variances in each operation.

The Waterfall Diagram that the Variance Handler generates provides a means for displaying cause and effect relationships in a Variance Model. The Waterfall Diagram presents a more detailed view of a process than does the IDL diagram. It displays the relationships between the variances and shows the cross-operation interactions.

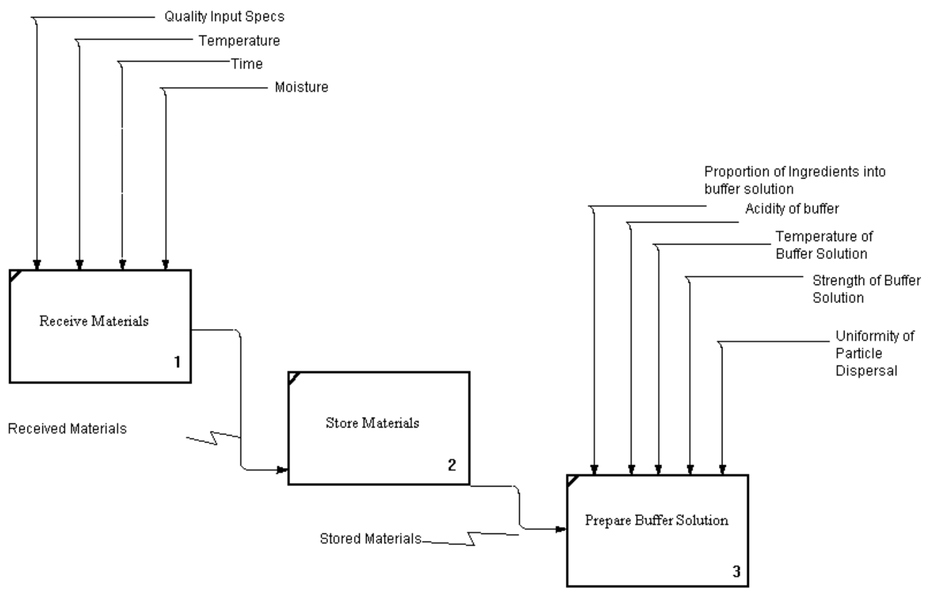
In a waterfall diagram the first column specifies what unit operation a variance exists in. The second column is the name of the unit operation. The third column "effect number" is a variance number which is in the sequence in which the variances are first observed. The fourth column "effect name" is a name of a variance. The remaining columns are marked "v1...vN" for variance "1...n. The "effect number" and "effect name" are preceded by effect because they denote variances that are affected by other variances upstream. The numbers in the columns marked v1...vn denote dependency relationships between variances.

A partial IDL Diagram and waterfall diagram view of a variance model is shown below. In this particular model in the first unit operation, there exist the 4 variances *Quality of Input Specs*, *Temperature*, *Time*, and *Moisture*. Both the IDL and the Waterfall diagram show this. Note that the waterfall diagram view provides even more detailed information as to how the variances affect each other. The waterfall diagram view states that the variance *Quality Input Specs* affects *Temperature*, *Time*, and *Moisture*. These variances all exist in the first operation therefore these interactions are inner process interactions.



Unit Operation Number	Unit Operation Name	Effect Number	Effect Name	v1	v2	v3	v4
1	Receive Materials	1	Quality Input Specs	1	-		
		2	Temperature		2	-	
		3	Time		2	3	-
		4	Moisture		2	3	4

The Waterfall diagram can also tell us about cross process interactions. The next picture presents a wider view of the above variance model.



Unit Operation Number	Unit Operation Name	Effect Number	Effect Name	v1	v2	v3	v4	v5
1	Receive Materials	1	Quality Input Specs	1	-			
		2	Temperature		2	-		
		3	Time		2	3	-	
		4	Moisture		2	3	4	-
3	Prepare Buffer Solution	5	Proportion of Ingredients into buffer solution	1				5
		6	Acidity of buffer					5
		7	Temperature of Buffer Solution	1				
		8	Uniformity of Particle Dispersal	1				5
		9	Strength of Buffer Solution	1		3		

inner process interactions

cross process interactions

This diagram shows three unit operations. Note that in the IDL diagram, only operation 1 and operation 3 have variances. The waterfall view below therefore has no listing for operation 2.

This more extensive waterfall diagram view demonstrates cross-process (cross-operation) interactions.

Process Replication

Although this section is not strictly part of the Cross-Process Integration subject matter, we explore it here because it does relate more to issues of process integration, than the other topics in the current version. (When the next booklet version comes out, this section will probably move to it rightful place.)

A frequently occurring problem is that of deploying the same architectural component in many different settings, e.g. many different geographical regions, or many different countries, or many different customer locations.

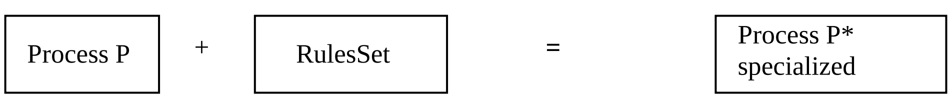
Each venue of deployment poses a set of specialized process circumstances, almost always resulting requirement changes from the original base ProcessP1 to a new ProcessP2, e.g. special handling for specific kinds of transactions.

For some software developers the most straightforward solution is to replicate the software system in each instance, and then modify the underlying source code to generate a new version: This yields a version for Massachusetts, a version for Texas, a version for California, and so on.

Our strong advice is not to tailor a large number of different versions for each slightly different process: there's no reason to replicate the code, other than for reasons of inducing separation in highly-specialized security and control environments. It is far easier to run the same code, but to achieve the separation of conditions via other means, such as a specialized rules base.

In the case of multiple airline subsidiaries of AMR Corporation, (American, Canadian Air, and American Eagle), it is often necessary to invoke a carrier-specific rule for purposes of error handling, or exception processing.

This is what we call *Process Specialization*. In this approach a given ProcessP binds to a RuleSet that contains all the necessary specializations, thus yielding a new ProcessP*.



The same sort of specializations occur with multiple companies in a conglomerate, e.g., some will use inventory valuation with LIFO rules, others will use traditional FIFO valuation models.

So long as the exception conditions are not too numerous, a straightforward rules base will suffice to cover fairly significant differences, even if the differences are many in number.

Sources of Specialization

Specialization is an important aspect of cross-process problem solving, because differences in customers, products, markets drive differences in the processes that serve them. If one attempts to solve each process variation according to its apparent distinct and different properties, the usual outcome is enormous complexity, and potential geometric increase in the number of cases one has to deal with.

With proper specialization and adaptation approaches, differences that on the surface appear unique, vanish almost entirely below the level of the user interface interactions.

Each business process requires a component-set. These components are not wildly different from one specialization to another; in fact, they often resolve into a small number of powerful, recurrent abstractions that we call PATTERNS. Some of the component sets are like "infrastructure', that is, they are relatively fixed elements of the environment, each one having specific functional (transformational, operational) capability. When this happens, these components become reusable. One need not continually reinvent the process "wheel". One size will fit all, provided the proper mechanisms for pattern recognition and reuse exist.