

Adaptive Policy Administration

*12 Ways to Know if a
System Is Truly Adaptive*



This edition of the Adaptive Policy Administration Systems white paper is being circulated to insurance and IT industry leaders to initiate a dialogue about adaptive systems.

Our goal is to articulate what makes a policy administration system flexible in ways that provide business value. We believe establishing such clarity will help insurers as well as system vendors like Adaptik.

In this edition, you will find updated examples based on the feedback we have received from readers of the previous editions of this paper as well as industry leaders and professionals with whom we have had the pleasure to share our ideas.

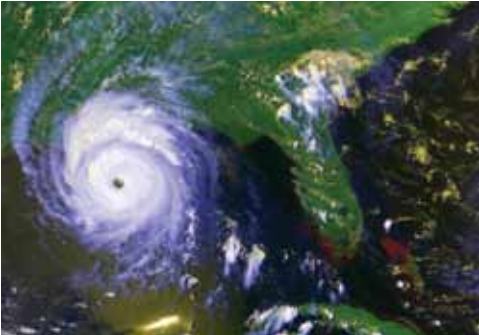
Please feel free to share this paper with your colleagues. We appreciate any feedback you would like to provide. The feedback received will be used to extend and refine future editions. Please send comments to Laura Ford at lford@adaptik.com.

Adaptik is an insurance software company based in Bethlehem, PA. Our flagship product PolicyWriter is a highly adaptive policy administration system. More information about Adaptik is available at www.adaptik.com.



A REAL-WORLD SCENARIO

Almost 20 years ago, Hurricane Andrew roared ashore in Florida, inflicting a horrible toll in deaths, injuries and property damage.



Hurricane Andrew set an historic precedent that introduced new requirements for insurance systems.

However, it doesn't take a catastrophe to force change in the insurance market. Changes to system requirements are identified on a daily basis in response to shifts in business, technology and regulations.

On August 24, 1992, approximately 25,000 homes were destroyed and more than 100,000 were damaged. All told, the loss to common insured private property was more than \$15 billion dollars, resulting in an insurance crisis in the state.

Reaction to the crisis came quickly from the Florida Legislature and the Department of Insurance. Several new regulations were enacted, including one that sounded fairly simple. After Andrew, insurers would be required to offer a discount for homes that were equipped with hurricane shutters. From a business standpoint, all that was needed was to begin asking customers if they had shutters on their homes and to make an adjustment to the premium based on the answer.

That seemingly simple business change turned out to be a major obstacle for the policy administration systems that were expected to support this new regulation. The addition of a new question and rating factor presented a big problem to these inflexible systems. One major insurer spent 14 staff months to implement this "simple" change.

Hurricane Andrew was a dramatic event that resulted in new requirements for insurance systems. However, it doesn't take a major hurricane to force change. In fact, changes to system requirements are identified on a daily basis due to shifts in business, technology and regulations. Many of these go unaddressed because there is not enough time or money to make the needed adjustments to existing systems. Many business users won't even bother to ask for system modifications because they know to expect a negative response from their backlogged IT staff or system vendor.

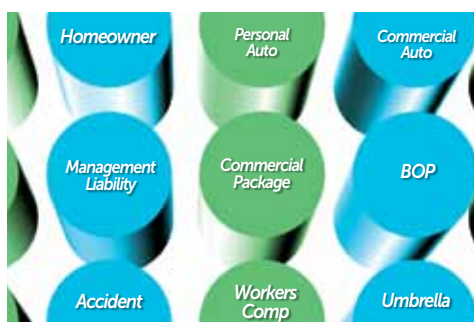
Today, nearly 20 years after Hurricane Andrew—and only four years after Katrina—most policy administration systems are in no better shape. Existing problems are magnified by the stepped-up rate of change fueled by advances in technology, industry consolidation, dismantling of regulatory barriers and new product offerings. Due to ever-escalating expenses, insurers are turning to cost-containment measures such as outsourcing.



A COSTLY LEGACY

Policy administration systems in use today are notoriously costly and difficult to change.

Some insurance companies spend in excess of one hundred million dollars a year enhancing and maintaining their systems, and yet one business executive complained that the real costs are in the business opportunities that are lost because of system inadequacies.



Property and casualty insurers have been forced to develop dozens of separate stand-alone or “stovepipe” systems, one for each new product offering.

The underlying problem is that these systems were designed to support a specific set of products and functions within a rigid framework. It is often not possible to extend these systems to support newly introduced insurance products. Their complexity increases with each change such that, after a few years, even small changes become difficult to make without breaking something else within the system.

As a result, property and casualty insurers have been forced to develop dozens of separate stand-alone or “stovepipe” systems, one for each new product offering. These systems significantly weaken an insurer’s ability to compete. Customers, agents, and company staff who interact with these disparate systems often complain about their inconsistencies and the complexity of conducting business and introducing new insurance products.

Inflexible vs. Adaptive Systems

Against this background of inflexible systems and escalating costs, there has been a growing realization that, to truly be effective, policy administration systems must be adaptive. Consequently, there is now a heightened interest in systems that can not only withstand change but also accommodate it rapidly without reprogramming or increasing the overall complexity of the system.

Adaptability is important to policy administration not only because of the need to accommodate change but also to support the housing of multiple insurance products within a single system. Legacy “stovepipe” systems were created out of the inability of traditional systems to simultaneously support multiple insurance products that often have different data capture, processing actions, workflow and even terminology requirements. The same powerful adaptive features that help accommodate change for an insurance product also make it possible to support multiple insurance products within one system without forcing a “one size fits all” approach.



WHAT IS AN ADAPTIVE SYSTEM?

Adaptive systems are broadly defined as “systems that can be changed through configuration rather than programming.”

Adaptive Terminology

Adaptability

Indicates how easily the system will support business and technical requirements that aren't yet known. If a system is fully adaptive, it can easily be changed to meet existing requirements and those of the future. The two main characteristics of an adaptive system are **configurability** and **extensibility**.

Configurability

Ability of a system to accommodate certain changes to its contents, behavior and appearance without reprogramming. Configuration changes are not made by programmers, but by authorized system users by means of user-friendly screens and dialogues.

Extensibility

Ability of a system to accommodate certain changes easily without requiring structural alterations. Extensible systems anticipate future changes so that completely new elements can be introduced with no increase in system complexity.

Thanks to such a vague definition, most software vendors claim that their systems are indeed adaptive. But do their software packages really offer critical flexibility in the right places and to the degree necessary to be beneficial?

Providing a few configurable parameters is certainly not sufficient for a system to face the real-world challenges of business change. A full-fledged adaptive system provides easy-to-use configuration facilities that allow system content and behavior to be changed without the need for reprogramming that often involves costly resource- and time-intensive product development cycles. Changes that can be accommodated in this way are considered to be configurable. To earn the badge of “Adaptive System,” most changes must be accommodated by configuration instead of programming.

On the other hand, it is neither possible nor desirable to accommodate all change through configuration. To avoid unnecessary expense or system complexity, certain structural elements that change infrequently may not be made configurable. And while some programming intervention is acceptable, adaptive systems must have built-in extensibility features to allow such changes to be made easily.

The Fallacy of the “Universal” System

Don't assume that an adaptive system must be a generic or universal system that can be used to operate a shipping business just as effectively as an insurance company. Universal systems that try to be everything for everyone are overly complex and ineffective because they attempt to support a variety of business models. In contrast, the most successful adaptive systems focus on one particular industry or industry segment.

This focus enables designers of an adaptive system to anticipate the kinds of change that will be encountered by that business. As a result, flexibility can be targeted to where it will yield business benefit and avoided where it will be superfluous and only add complexity.

An industry focus also helps the adaptive system designer devise an effective configuration facility. Effective design requires knowing the specifics of each targeted change, how often it occurs and the typical skills of the staff who will be administering the system. In addition, configuration can be made intuitive and easy to learn through the use of industry-specific terms and concepts that are instantly familiar.



EVALUATING POLICY ADMINISTRATION SYSTEMS

Evaluation of traditional policy administration must include adaptability.

Evaluation of modern policy administration systems must look at *adaptability* in addition to *functionality* and *technical fit*.

Adaptability is the measure of how well a system will support business and technical requirements that are not yet known.

Functionality is how closely the system's features meet the current requirements. Technical Fit is how well the system performs on the platforms and environment where it will be deployed.

But with adaptive systems, there is a third dimension to consider—Adaptability. Adaptability indicates how easily the system will support business and technical requirements that aren't yet known. In fact, the pervasiveness of change in policy administration makes this third dimension more important than functionality in an evaluation. If a system is fully adaptive, it can easily be changed to meet existing requirements and those of the future. A non-adaptive system that completely meets existing requirements, but can't easily be changed will soon be another burdensome legacy system.

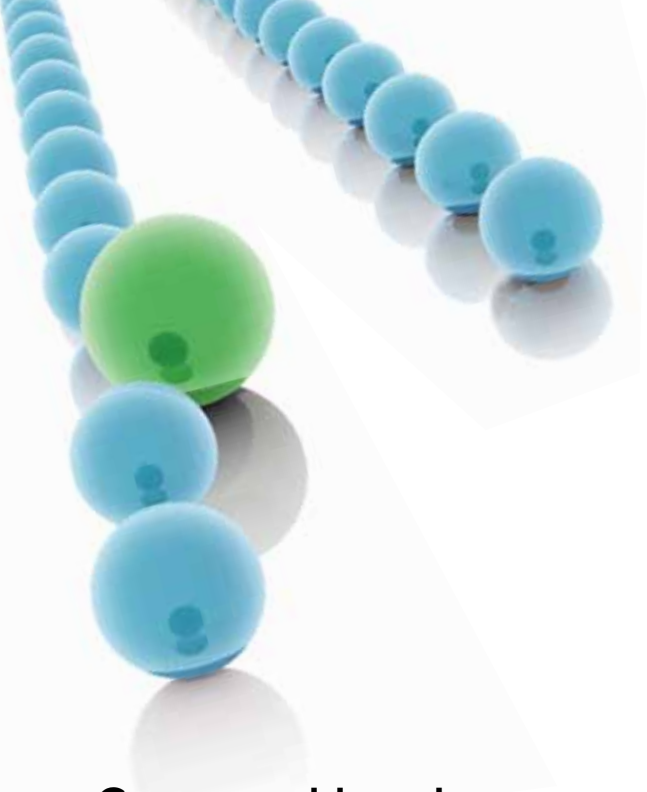
Determining which systems are truly adaptive can be tricky—there is often a quagmire of marketing hype to wade through. Some vendors claim to have adaptive applications after adding a few user-maintained parameters on top of a traditionally designed system.

Fortunately, there are some fairly simple tests for determining a system's adaptive quotient. Adaptive systems must possess a comprehensive set of capabilities to deal adequately with change. With some systematic probing, one can detect the presence or absence of these capabilities and evaluate the degree to which a system is configurable and extensible.

12 CHARACTERISTICS OF AN ADAPTIVE SYSTEM

1. Flexible Data Architecture
2. Adaptable Functionality & Appearance
3. Adaptable Product Composition
4. Adaptable Reference Information
5. Dependencies
6. Variations
7. Configuration Sharing
8. Configuration Versioning
9. Rule Reapplication
10. Out-of-Sequence Transactions
11. Adaptable Interfaces
12. Rule Engine Integration





12 Characteristics of an Adaptive System

These characteristics are all essential to a truly adaptive policy administration system. Use these characteristics as guidelines for determining whether a system is sufficiently configurable and extensible.

Can you add or change stored data items without database or programming changes?

Adaptive systems avoid system rigidity by completely decoupling the data schemes for information content and physical storage. All content information is completely configurable, meaning that new data items can be introduced, fully specified and available within minutes—without programming or database changes.

1. Flexible Data Architecture

In traditional systems, data elements correspond to columns in database tables. Therefore, to add or change a data element, the underlying database must be changed along with the programs that utilize that item.

Adaptive systems avoid this root cause of system rigidity by completely decoupling the data schemas for information content and physical storage and making all information content configurable. This way, new data items can be introduced into the system and related to other data items through a configuration facility within minutes and without the need to make programming or database changes.

The overwhelming advantages of decoupling the data schemas for information content and physical format are well known to all who are familiar with XML which has become a universal data communication standard within a few short years. Adaptive systems extend these advantages into data capture, manipulation and storage and readily provide all policy data in XML format to external systems. As new data items are introduced, they automatically appear in the XML stream without the need for any programming changes.



2. Adaptable Functionality & Appearance

Can your system's look and behavior be changed by users—rather than programmers?

An adaptive system's configuration facility makes it possible to control placement of elements on a page, conditional processing of those elements, and even determine how—and to whom—these elements will be displayed.

Adaptive systems make it possible to adjust such attributes on the fly without requiring database or programming changes. They're also extensible—allowing the introduction of new web pages and processing logic with little or no programming.

With adaptive systems, the appearance and functionality of a system must be easily varied. Users should be able to configure the placement of elements on each web page and adjust the numerous properties that control appearance and behavior of the user interface.

Other configurable parameters should include those that control web page navigation, the available commands and the points at which procedures such as assessment of underwriting rules and rating are automatically triggered. Adaptive systems should also be easily extensible to introduce whole new web pages and procedures within hours or days, with little or no programming.

For example, an adaptive system would allow an authorized user to place questions in a particular spot on the desired page. This user would also be able to configure other properties such as whether the answer is given via a dropdown or radio button control and to specify the literal displayed beside the control.

User Configuration

Users should be able to configure system appearance and behavior. For example:

- Adding new data elements and specifying whether they are required, the available answer options, the default values and the display characteristics (e.g., screen control, user access, hidden, protected, font color & style, etc.)
- Data formatting and validation, including prefilling, deriving and defaulting data values from other elements previously captured on the policy
- Conditional, rules-based processing for many aspects of the system, including product/coverage availability, application navigation, data capture and answer options
- Triggering additional system processing that should occur when policy data is entered or altered, such as rule reapplication or integration with external systems
- Leveraging and sharing data elements across multiple products



Can you easily introduce and maintain different insurance products with unique structures?

In addition to providing flexible data definition for policies, an adaptive system's configuration facility also makes it possible to configure the data that is used to describe the insurance products themselves.

In other words, product definition is not limited to a particular data model that has been hard-wired into the system.

3. Adaptable Product Composition

Insurance products consist of various product entities such as lines of business, coverages and forms. An adaptive system's configuration facility should make it possible to compose multiple insurance products including package and monoline products using these product entities.

Look for all the product entities to be configurable such that instances of them can be added and removed easily (e.g., adding and deleting specific coverages). Each product entity may have a number of attributes. For example, a coverage may be required or optional; it may apply at the policy level, by location, by vehicle or by some other criteria; and, it may or may not require itemizing the items to be covered. The configuration facility should also allow changing the attribute values of product entities.

In addition, the need to add new attributes to existing product entities arises frequently. For example, suppose there is a new business requirement for the system to automatically pre-select some coverages whenever a policy quote is initiated. To achieve this, a new configurable coverage attribute such as "auto-select for quote" would need to be introduced. Adaptive systems must be able to add such attributes on the fly, without requiring database or programming changes.

A Multi-Tiered Meta Data Architecture

Sometimes, as different products and functionality are introduced into the system, entirely new product entities need to be introduced and related to the existing entities. Suppose there is a new requirement to define applicable reinsurance agreements as part of each product. It should be easy to add a new configurable product entity (we may name it "Reinsurance Agreement") with a number of attributes that govern the applicability and execution of reinsurance for each product.

Introducing new product entities and attributes without having to make programming changes requires a multi-tiered meta data architecture that enables the data flow content to vary dynamically throughout the system. This ability sets adaptive systems apart from applications with rudimentary service-based architectures.



4. Adaptable Reference Information

Does your system allow you to declare and populate reference entities?

Adaptive systems provide the means to define new reference entities in the data model that can be added, removed and related to one another easily through configuration.

Adaptive systems can also access reference information that resides in corporate databases to avoid dual maintenance of the information, such as zip+four data, county of residence and so on.

Policy administration systems make considerable use of information about reference entities such as countries, states, counties, territories and writing companies. Adaptive systems should be able to access such information in corporate databases and provide the means to define new reference entities that can be added, removed and related to one another easily through configuration.

For example, the system should allow you to add a new entity named Territory and relate it to the State entity using the configuration tool. In addition, each reference entity may have a number of attributes—a Writing Company may have an Address, and a Territory may have a special Code that is recognized by a downstream system.

Adding New Reference Entities and Attributes

As in the case of product entities and attributes, the need often arises to add new reference entities and attributes. Take a case where, according to an entirely new business requirement, the system needs to automatically determine whether to use the effective date or the processing date of the policy, depending on the laws of each state. To achieve this, a new configurable attribute would be added to the State entity. Adaptive systems can introduce such attributes immediately through built-in extension features.

As another example, suppose there is a new business requirement to process multiple currencies. Adaptive systems must be easily extensible, to introduce Currency as a new reference entity that is fully configurable and allow system users to enter instances of currency (e.g., \$, ¥, £, etc.) and their associated attributes into the system from that point on.

Note that, while reference information may be shared across products, adaptive systems must provide the means to limit their utilization by product. For example, all 50 United States may exist as reference data, but a particular product may be configured to make use of only three states.



Can your system dynamically decide to collect different information based on context?

In an adaptive system, it is possible to alter the application appearance, behavior and business logic based on user authority or the values of one or more data elements. This eliminates the need to prompt the user for irrelevant or unnecessary information.

5. Dependencies

An essential feature of adaptive systems is the ability to configure each system element conditionally. In other words, it must be possible to alter the appearance, behavior and business logic based on user authority or the values of one or more data elements.

For example, it wouldn't make sense to ask the user to enter the name of an insured's spouse if they had just indicated that the policy applicant was unmarried. In this example, a simple dependency (*If Marriage Status = "Married," then prompt for Spouse Name*) is used as a means to avoid asking for information that is irrelevant or unnecessary based on the context of a particular policy.

Suppose that in Texas there is a requirement that the question "Is the occupancy only an office?" is not to be displayed on screen, but its answer is to be automatically stored as "Yes" for use in rating. Such a requirement would be handled with a dependency condition that is directed at the default value and display property of the particular question. This dependency would be: *If State = "Texas," set default value = "Yes" and display property = "Hide."*

It should be possible to control the properties of one data item based on the existing values of one or more other data items anywhere on the policy. Complex conditions involving several data items must also be supported utilizing Boolean operators such as AND, OR and NOT. For example, consider that the name of the spouse is only of interest in California. Then, we would have a more complex dependency (*If State = "California" AND Marriage Status = "Married," then prompt for "Spouse Name"*).

Look for the ability to vary most configurable properties using dependencies. Make sure that the condition for a dependency is not limited to information from the same page. Conditions should be able to refer to any previously entered information on the policy.



6. Variations

Most insurance products are replete with rules that are based on several permutations of one or more principal factors (e.g., State, Product, Coverage or Program Type). When designing a product, insurers will typically first specify what is standard for the product (e.g. Country-wide), then formulate exceptions based on these principal factors.

Most rules-based systems can accommodate the rules needed to process both standard and exception situations only through considerable rule complexity. Translating these requirements into dependent processing conditions for a typical rules-based system for each coverage and coverage attribute value is a complicated process. Because of this complexity, the process of interpreting requirements, building, testing and maintaining the rules can be both onerous and error-prone.

To appreciate this point, consider the following example—a Business Owner's Policy product whose coverages apply only for certain programs in certain states.

Can your system easily manage large numbers of rules that vary by multiple factors?

Adaptive systems provide a mechanism to capture all the coverages with their standard attribute values, and then allow variations to be made by principal factors (e.g., State and Program).

This approach has dramatic advantages over the usual tedious and error-prone method of using compound dependency conditions for each coverage and coverage attribute.

COVERAGE	DEPENDENCY LOGIC
Business Owner's Liability	If Product = Business Owner's THEN Available ELSE Not Available
Tenant's Fire Liability	If Product = Business Owner's THEN Available ELSE Not Available
Contractor's Liability (Painting Only)	If (Product = Business Owner's AND Program = Contractors) THEN Available ELSE Not Available
Contractor's Liability (Other than Painting)	If (Product = Business Owner's AND Program = Contractors) THEN Available ELSE Not Available
Hired Auto	If (Product = Business Owner's AND PROGRAM = Contractors AND State = TX) THEN Available ELSE Not Available
Non-owned Auto	If (Product = Business Owner's AND State <> TX) THEN Available ELSE Not Available
Employee Benefits	If (Product = Business Owner's AND Program <> Contractors) THEN Available ELSE Not Available
Additional Insured Vendors	If Product = Business Owner's THEN Available ELSE Not Available
Professional Liability	If (Product = Business Owner's AND Program <> Contractors) THEN Available ELSE Not Available
Utility Service Direct Damage	If (Product = Business Owner's AND STATE = [NJ OR NY OR DE]) THEN Available ELSE Not Available

To make the point even clearer, scan the dependency logic above—and try to decipher which coverages in Texas apply for the Contractor's program. Not easy, is it? In contrast, adaptive systems use the variation approach for handling rules that are based on permutations of principal factors.



In the context of our example, the configuration facility captures all the coverages with their country-wide attribute values and then variations from these rules are configured, reducing the configuration and testing complexity. This ability to easily modify the applicability of a coverage makes the adaptive system easy to use and the configuration effort much more manageable. Below is an example of the same rules configured in an adaptive system.

Adaptive systems manage the complexities of variations behind the scenes, providing an intuitive configuration user interface to manage rule variations.

COVERAGE	STANDARD SETTINGS	CONTRACTORS				OTHER THAN CONTRACTORS			
		NJ	NY	DE	TX	NJ	NY	DE	TX
Business Owner's Liability	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tenant's Fire Liability	✓	✓	✓	✓	✓	✓	✓	✓	✓
Contractor's Liability— Painting Only	✗	✓	✓	✓	✓	✗	✗	✗	✗
Contractor's Liability— Other than Painting	✗	✓	✓	✓	✓	✗	✗	✗	✗
Hired Auto	✗	✗	✗	✗	✓	✗	✗	✗	✗
Non-owned Auto	✓	✓	✓	✓	✗	✓	✓	✓	✗
Employee Benefits	✓	✗	✗	✗	✗	✓	✓	✓	✓
Additional Insured Vendors	✓	✓	✓	✓	✓	✓	✓	✓	✓
Professional Liability	✓	✗	✗	✗	✗	✓	✓	✓	✓
Utility Service Direct Damage	✗	✓	✓	✓	✗	✓	✓	✓	✗

✓ = Available ✗ = Not Available

Again, scan the variation logic above and decipher which coverages in Texas apply for the Contractor's program. This time it is intuitive.

This example has been kept simple for illustration purposes. In the real world of insurance product definition, there are likely more than two principal factors. These principal factors impact multiple coverage attributes—e.g. specific coverages may be mandatory in one state and optional in another. And finally, each principal factor will most certainly have many values—e.g. 30 programs and 50 states are not unrealistic. So it is easy to see how the complexity is compounded without a mechanism specifically designed to handle these types of rules.

Adaptive systems manage this complexity behind the scenes. They also provide an intuitive configuration user interface to configure and manage only the rule variations for a set of factors. This approach eliminates the significant amount of work that is introduced when the configuration for each state must be separate

Bottom line: This makes adaptive systems faster to market, easier to use and more manageable to maintain.



Can you configure a repackaged product only once and add variations when needed?

Adaptive systems must provide a mechanism for effective configuration sharing among products with common characteristics while still allowing such products to be configured differently where needed.

Can the system keep multiple rule versions over time and use each when appropriate?

Recognizing that each product goes through continuous change, an adaptive system must provide convenient and effective ways of creating new rule versions, testing them and migrating them into production.

7. Configuration Sharing

Often insurance products have overlapping parts. For example, Inland Marine may be offered as a monoline product and also included in a package product. Sometimes a package product may be modified into a new product to target a different market segment. In these cases, it is often desirable to share the configuration for the overlaps among products to avoid significant duplication of work.

Adaptive systems must provide a mechanism for effective configuration sharing among products with common characteristics while still allowing such products to be configured differently where needed.

For example, Directors & Officers Insurance is typically offered as a monoline product, but it is also often combined with other lines of business to make package products that are attractive to specific market segments such as private and publicly held corporations. An adaptive system makes it possible to define the D&O rules only once for both the monoline and package products. However, the system also allows customizations for each of the products involved, such as different limits for private and publicly held corporations.

8. Configuration Versioning

Insurance policies are legal contracts that insurers make with customers. As such, each policy is tied to the particular version of the rules that coincides with its effective or processing date. (Some states require the use of the date a policy is first created while other states mandate the use of the policy effective date for this purpose.) Therefore, as product rules are changed over time, the earlier rules can't simply be discarded. Policy administration systems must be able to keep past, present and future versions of the rules and automatically determine when to enforce which version.

Recognizing that each product goes through continual change, an adaptive system must provide convenient and effective ways of creating new rule versions, testing them and migrating them into production. Furthermore, configuration staff must be able to work simultaneously on multiple sets of rules with different effective dates. Therefore, adaptive systems must provide the functionality to automatically "merge" the changes from one rule set to another in effective date order, and also allow reordering of the rule sets if the effective dates are altered due to changing business needs.

When a new rule version is introduced, the system must make sure that the existing quotes and policies are not affected while providing the ability to easily "switch" individual quotes and policies to the new rule version when desired. Switching rule versions for an existing quote or policy is discussed below under "Rule Reapplication."



9. Rule Reapplication

Policy administration systems validate every piece of newly captured data against product rules. However, a change to policy data that impacts the rules for information already captured can cause policy data to become inconsistent with the rules. Adaptive systems must have the capability to automatically detect these cases and make the necessary adjustments without requiring the users to reenter data.

Can your system automatically adjust a policy to conform to product rules?

Changes that alter the applicable rules for a policy represent a special challenge for legacy systems.

In contrast, adaptive systems employ sophisticated features to detect whenever any policy data item becomes inconsistent with product rules and automatically make the necessary adjustments.

This capability known as “Rule Reapplication” is now taking center stage as a fundamental building block for most high-order system behavior sought by insurers to drive competitive differentiation.

To understand the need for rule reapplication, consider a commercial auto policy where Use Class of “Other” is not permitted when Vehicle Type is “Truck.” Changing Vehicle Type to “Truck” within an existing quote or policy will cause the data to become inconsistent with the rules if Use Class happens to be “Other.” A simple change like this could also impact the coverages offered, available limit values, and the required underwriting information. Most systems cannot deal with the complexity of these resulting inconsistencies and, therefore, do not allow such changes. A system equipped with rule reapplication, however, can automatically adjust the related policy data so it conforms to the rules based on the policy change.

A fundamental building block

A special case of rule reapplication becomes necessary whenever Policy Effective Date is changed after policy data has been captured and saved, typically during new business and renewal transactions. The changed date may require an entirely new set of rules to be applied against existing policy data.

When the Policy Effective Date needs to be revised on a new business application, insurers have had no choice but to require users to discard the first application and begin a new one. However for renewals, re-entering policy data is not viable for adjusting the effective date to reflect the new policy term. In this situation, the rules in force for the new term are almost always different than those during the previous term. Without rule reapplication, the system must be specifically programmed (a.k.a. hard-coded) for each possible inconsistency to make the required adjustments. Given the number of changes made each year for each jurisdiction, it is easy to see how the cost and complexity of maintaining existing policy administration systems increases over time.

Beyond providing a generalized solution to this important problem, rule reapplication is a fundamental building block for key high order policy administration behavior such as insurance product evolution, book-of-business acquisition, annual follow-up on lost business, comprehensive support for out-of-sequence transactions (see below) and automatically adapting policies to changing customer needs. Applications that cannot reapply rules without resorting to hard coding are predestined to fail as adaptive systems.



10. Out-of-Sequence Transactions

Can you process all types of out-of-sequence transactions without any limitations?

Adaptive policy administration systems are uniquely positioned to process out-of-sequence transactions automatically, without limitations, and without requiring programming changes even when multiple policy terms are involved. This is principally due to their metadata foundations and their ability to easily and quickly reapply rules.

Unlike rule reapplication, out-of-sequence transactions have long been included in short lists of must-have functions for a policy administration system. The basic requirements are most often associated with a policy change that must be made after other transactions with later effective dates have already been processed on the policy. However, the need for out-of-sequence transactions arises in a number of other scenarios where the changes being made impact other transactions that are still in process and/or possibly involve multiple policy terms.

For example, suppose a policy is in the process of being renewed and a renewal quote has been created with the proposed changes several weeks before the renewal date. At this point, the customer calls and wishes to add a coverage to the policy effective immediately. With out-of-sequence transaction capabilities, the system would be able to add the coverage to the policy for the current term and, if desired, also automatically add it to the renewal quote.

Processing Out-of-Sequence Transactions

Out-of-sequence transactions are challenging because they need to address the contradictions that result from “time travel.” For example, consider a policy change that increases the limit for a coverage. If an out-of-sequence change jumps before it and deletes the coverage in question, increasing the limit of a deleted coverage no longer makes sense.

While most system vendors claim to support out-of-sequence transactions, it is difficult to find one that provides the full automation required without various restrictions. Furthermore, when rule reapplication (Section 9) is not available, the programming that deals with the time travel contradictions is hard-coded and must often be reworked and retested as product rules change.

With their meta-data foundations and the generalized ability to reapply rules, adaptive policy administration systems are uniquely positioned to process out-of-sequence transactions automatically, without limitations, and without requiring programming changes even when multiple sets of product rules are involved.



11. Adaptable Interfaces

No system is an island. Adaptive systems are designed to thrive in an ever-changing environment with evolving interfaces to both internal and external systems. While traditional systems often require substantial coding changes to incorporate a new or altered interface, adaptive systems anticipate and accommodate such change through configuration and extension.

Can interfaces with outside systems be configured rather than programmed?

With an adaptive system, interfaces are rapidly configured as needed. This flexibility means that an adaptive system can be implemented in successive steps based on the needs of the business, dramatically reducing the risks and costs involved with integrating new or existing systems or replacing legacy applications. In fact, adaptive systems are designed so that they can easily evolve as business needs and requirements change.

Creating a generalized framework for data exchange

The key to adaptive interfacing is the ability to invoke a variety of external modules (e.g., web services, EJBs, etc.) precisely when they are needed, provide them with the required information, and make use of the information that is returned—all without the need to reprogram the system. In addition, when communicating with other systems, the system must be able to conform to open standards (e.g., ACORD) and to easily adapt as existing standards evolve and new standards are developed. For policy administration, all of these interface capabilities must be permitted to vary easily by insurance product to accommodate the specific product needs and different governing bodies involved.

Leveraging specialized third-party information

With the increasing use of web services, many types of up-to-date specialized information—address validation, credit scoring, driving records—are becoming available at a nominal cost. Adaptive systems easily take advantage of these and other business-to-business services by incorporating them into the business process. Most importantly, this is done at the point where the value is maximized—not where it is most convenient for the system.

Accepting data from external systems easily

Adaptable interfacing makes it possible to load quotes and policies from other systems such as agency management systems. Adaptive systems can accept such data and apply the same edit and validation rules that are used for direct input from web pages. This ability greatly simplifies conversions from old systems and finally brings configurability to conversion tasks.

Managing implementation risk incrementally

The flexibility of an adaptive interface is that risky “big bang” implementations are no longer required. Interfaces may be configured as needed. One interface can easily be replaced by another. In this way, it becomes much easier to implement system solutions in small successive steps. Furthermore, interfaces configured with an adaptive system are compatible with future versions of the system software, allowing new functions to be introduced without a need to rework the integration. An adaptive system continues to evolve along with the environment in which it is installed.



12. Rule Engine Integration

Can your system utilize any rule engine with little or no programming?

An adaptive policy administration system can easily take advantage of your existing rule engine investments. Rules can be accessed as needed, eliminating the need for dual maintenance or migration of existing rules into the new system.

This plug-and-play capability is critical because the software industry is rapidly moving toward the use of vendor-provided multi-industry rule engines that can be attached to a variety of business applications.

It may seem counterintuitive, but the most effective adaptive systems aren't those that attempt to house all the business rules. Paramount for adaptability is the ability to incorporate stateless external rule processing components that are commonly called rule engines.

For example, an adaptive system should be able to utilize a customer's proprietary forms selection engine alongside an underwriting engine from one vendor and a rating engine from another without any difficulty. Furthermore, the system should be able to use different rule engines by insurance product. If the system comes with a built-in rule engine, it should be replaceable by third-party software when preferred.

This plug-and-play capability is desirable because the software industry is rapidly moving toward the use of vendor-provided multi-industry rule engines that can be attached to a variety of business applications. These rule engines can quickly evaluate a business item such as an insurance quote request or loan application against a pre-defined set of conditions and return the results for further processing. Several increasingly sophisticated rule engines are already available on the market, and they are becoming very reasonably priced thanks to their relatively wide customer base. Most important, these systems are often themselves adaptive—allowing the conditions to be varied easily without requiring programming changes.

Insurers must be able to pick and choose the best-of-breed from these rules engines and know that their policy administration system will be able to easily meld them into one seamless and configurable solution.

Look for an adaptive policy administration system that is capable of handling a changing collection of rule engines rather than one that attempts to restrict the choices to a single vendor package.



Conclusion

Adaptive systems are fundamentally changing the economics of policy administration by altering the traditional time frames and cost structures for system ownership.

In addition, insurers stand to gain significant competitive advantages by leveraging the speed and flexibility of these systems in introducing a wide variety of customized insurance products to serve domestic and international niche markets.

Therefore, new policy administration software solutions must be highly adaptive. It is no longer sufficient to select software that simply matches a finite list of features—adaptability has become the critical evaluation criterion. Insurers are finding it necessary to opt for systems that are capable of being configured to meet the requirements of today in addition to the yet unknown requirements of the future. There is no other viable option for becoming light on your feet while dramatically scaling back unsustainable IT spending. Pick a robust, industry-focused adaptive system, and you may never have to search for a new policy administration system again.





ADAPTIK CORPORATION

1495 Valley Center Parkway

Suite 120 | Bethlehem, PA 18017

(800) 306-1062 | (610) 814-4845

info@adaptik.com | www.adaptik.com

