Business Rules are from Mars & Processes from Venus

Introduction

Over the years, businesses have sought ways of making their systems more extensible and flexible. In support of that aim, two fundamental different approaches emerged for software developers to parameterize the way that systems operate, allowing their configuration and adaptation over time; without necessarily requiring the rewriting of software.

For many years, the two camps operated relatively independently of each other, pushing the envelope of what was possible with their respective approaches. As a result, each developed its own of set of firm followers and vocal evangelists, singing the praises of their respective approaches while failing to fully understand the advantages of the other methodology. The two approaches we now know as Business Process Management (BPM) and Business Rules (BR). More recently, driven by the need for greater agility combined with enterprise-wide efficiency, many firms have started to meld the two approaches. Indeed, wholesale evolution in the BPM market is occurring as enterprises look for BPM Suites with Business Rules capabilities as a core component—not an incidental add on.

The reality is that these two approaches are intimately related—just different sides of the same coin. On the one hand, we could regard a process as the way in which a rule is implemented (for example where a rule might be some overarching business policy). On the other hand, a rule usually makes sense when interpreted in the context of decision making within a process of some sort.

While they are just two sides of the same coin, they are rarely developed and deployed as such. That is, until BPM systems emerged with Business Rules as a core and fundamental unit. This paper explores the benefits and advantages of this new phase in BPM, with the two approaches effectively combined. It also considers the different tactics used by vendors in an effort to achieve this fusion. The core question explored is whether “Loose Coupling” of BPM and Business Rules is better, or worse, than approaches that meld the two paradigms in a common environment, creating a superset of both rules and processes.

The Power of Rules Enabled Processes

Using business rules within process management technology, an organization can develop a set of business capabilities that directly underpin and drive its efforts to streamline operations, enhance customer service and bring new products to market. This approach enables the development of a valuable corporate asset, allowing the firm to more rapidly develop and optimize products and services that differentiate its offerings, while delivering a competitive edge over competitors. Rules empowered process management more readily supports the entire scope of enterprise activities. Collectively they are better able to support the unique needs of each department and division (even down to the level of individual customers); yet still enable consistent service standards and compliance with organizational directives or government regulations.

Well designed business processes can ensure efficient operations—they are the “how” of business. Using a process engine to drive work through the business has obvious benefits.

But the problem with procedural definitions is that they can become extremely complex as people attempt to capture all the logic related to the business area. Further, from a management point of view, it becomes difficult to apply consistent policies across sets of processes. A change in policy will necessitate redevelopment of each process.

On the other hand, business rules tend to reflect over-arching business policies, goals or strategies. Business rules are usually comprised of sets of declarative statements and constraints that can then predicate certain actions or goals. They are the “what”, rather than the “how” of business operations.

Treating the two as complementary approaches and leveraging the best aspects of each is far more effective than traditional approaches. By effectively melding processes and rules together into a truly common environment, it is possible to drive new kinds of optimization across a number of fronts:

- Simplification—integrated together, the mix of process and rules functionality enables far more flexible procedures than is normally possible with a purely BPM-based system. For example, in a Global 10 firm, using a combined business process and business rules environment, a single sales management process was able to handle 60 different products across 30 different operating companies.
- Speed of Response—applied intelligently, they facilitate the development of rapid business response mechanisms. By exploring different potential scenarios (for example, changes in interest rates, hurricane impacting the Gulf Coast, sharp increase/decrease in the price of oil, etc.), it is possible to develop robust process and rule combinations that are already optimized. Should that circumstance occur, the organization could turn on a dime to immediately deploy an optimal approach, ensuring efficient use of resources and/or rapid turn around of work.
- Analytics—integrated business rules capabilities can both monitor and drive optimization; effectively using the capabilities of the technology itself to ensure that operations are monitored appropriately (review and oversight policies observed) and, where suitable, invoke corrective action. This sort of thinking makes it possible to develop smart Service Level Agreements (SLAs); with built-in monitoring and escalation facilities, applying statistical sampling, and process cost calculation to underpin real-time optimization.
- Rapid System Development—together they enable firms to develop new products and services far more quickly than would otherwise be the case. In a study, undertaken by independent research company Strategic Focus, of Malpitas, CA, it took 38% less time to build, deploy and test applications with a combined process and business rules environment than it did with a modern Java development environment. Further, it then took 58% less time to change the completed application.

  Moreover, if approached correctly, it is possible to use the combined environment itself to support application development and deployment directly. The BPM project is itself wrapped and driven by the engine, ensuring rapid project completion and integrity of the developed application. In turn, this allows better control over how processes and rules are developed and deployed and, if need be, ensuring that business users themselves are the ones given responsibility for governing how their work environment operates.
- Controlled Evolution—it is possible to deploy specialized versions (or updates) of process and rule enabled applications to a specific set of users, without leaving the production environment. Effectively, this permits new functionality to be tested on live data, with real users, before rolling out more widely. This controlled deployment

capability supports the sorts of experimentation that is at the heart of institutional improvement and optimization programs. Think of it as an opportunity to implement the Toyota Production System within the context of a modern customer service oriented organization.

- Granularity—firms can also consider supporting complex sets of relationships that transcend organizational, national, and regional boundaries by resolving the correct process and rule set based on the context of the case in hand. Indeed, it then becomes possible to support highly granular business problems and market segments (even down to individual customers).
- Proactive Customer Engagement—in customer service situations, rules and processes combine very effectively to guide Customer Service Representatives (CSRs) with context-specific advice to help them anticipate and respond appropriately to individual customer requests. This facilitates the cross-sell/up-sell of offers, driving sales and enabling top line growth.
- Exception Handling—it is when things go wrong that business weaknesses are exposed. This is especially a problem in customer-centric/customer-facing processes. Given the ability to automate standard procedures, exception handling is a primary driver of business cost, service differentiation, customer perceptions of the organization, and overall business performance. So the ability to adapt and evolve the operating procedures of the firm as it responds to these exceptions is critical. Using rules and processes together effectively allows exceptions to become the building blocks for better customer service, driving continuous improvement. Indeed, it allows the firm to rapidly deploy a core application that meets the needs of the vast majority of cases, safe in the knowledge that exceptions can be handled at runtime.
- Multi-Channel Relationships—processes and rules combine to enable more effective multi-channel operations, ensuring the delivery of a consistent customer experience regardless of the access mechanism; via the Web, telephony systems or even mobile devices such as cell-phones and PDAs.
- Compliance—it is also possible to ensure that the audit trail captures the context of decisions, storing the version of the rule used alongside the process case specific data. Indeed, system can deliver compliance ready reports, allowing managers and auditors to browse the entire event cycle of a case along with all of factors that affected decision making. Furthermore, in the financial services industry, sophisticated rules capabilities are needed to influence the way work is assigned to employees, based on the complexity of the case in hand, employee skills, training, and experience or quality profile.

This list outlines the broad opportunity space for evolving BPM and Business Rules environments—where the two merge into a holistic whole, rather than as distinct techniques that merely borrow from each other. Products that deliver the most effective combination of process and rules are now setting the gold standard of business optimization potential. They reflect the current state of the art in the evolution of business process management.

When looking at the products themselves, there is a wide spectrum of product functionality delivering support for some, or all, of these capabilities. While many vendors will argue their ability to support most of these optimization scenarios, it is a question of degree and efficacy. Some will require careful design and implementation, while others will provide direct product functionality out-of-the-box that satisfies the need. It is a question of how difficult such functionality is to implement and maintain over time.

Fundamentally Different Technological Approaches

Given the different histories of the BPM and Business Rules communities we find a number of distinct approaches to merging the two domains.

Traditional BPM Systems Development

With BPM systems, the emphasis is on definition and deployment of procedural models that are used to drive work through a sequence of tasks. The process engine manages the state of individual work items, in accordance with defined models. The theory is that when business needs change (as they invariably do), the model is changed without requiring additional software development.

In one way or another, all BPMS products allow conditional routing—procedural conditions that are embedded in the process model itself. However, it is inappropriate to categorize these conditions and filters as "business rules" although many vendors refer to them in this fashion. If the condition evaluates to true, the case of work follows that path in the procedure (potentially creating a fork or parallel path).

Conditions are based on the variables associated with the process definition. Every BPMS system enables the creation of a set of placeholders (the variables) that are populated with information as the case moves through the process (either programmatically from third party applications or via direct user entry). The data is usually stored in RDBMS applications that are mapped using the process modeling environment. Subtly different, the object model of the BPMS is usually fixed and cannot be changed (anything not directly supported by that object model is deemed to be part of an external application).

Effectively, the variables enable the process engine to manage contextual information of individual cases. Data gathered at one step is reused at another—either in presentation to workers or customers via on-screen forms or used to invoke third party applications. It is also used to support automated routing logic. For example, if the Purchase Order is over $5,000, then route it to the Purchasing Manager for approval, if it is over $20,000 then send it to the CFO. This sort of simple condition is represented graphically in the process model.

Although it is possible to represent all potential paths through a moderately complex process using conditional routing logic, the resulting process map is often almost completely unintelligible. As the complexity increases, so it becomes ever more difficult to represent the necessary logic clearly. Moreover, each condition is buried within the process model itself. If the condition needs to change, say because the threshold level of the Purchase Order is increased, then the process model must be changed and re-deployed.

When it comes to routing work to the correct worker (to handle a specific piece of work based on the context of the case and the skills and availability of employees), virtually all BPM products rely on a simple role structure. A few vendors have developed bespoke functionality in this area, but most of them struggle to provide an effective solution. Of course, the conditional logic described above could, in some simple situations, support this type of functionality. But the reality is that the problem is far more complex and requires sophisticated business rules support.

Stand Alone Business Rules

With an approach based on Business Rules, a repository of rule sets is developed and then used to support decision making, policy enforcement and data transformation (potentially in any number of business applications, systems or processes). These may be developed in a number of ways, but end up as sets of IF-THEN-ELSE statements that are evaluated at runtime by the Business Rules Engine (BRE) based on the variables passed to it. The BRE retrieves the identified business rule and executes it, returning the result to the calling application. Initially, BREs were used to support decision making in relatively static systems (without a strong process orientation), enabling the development of more flexible applications.

The problem is that while an organization can build a series of bespoke applications that leverage the developed business rules, it is the slowest and highest risk approach. Inevitably, much of the functionality of those bespoke applications will have processes embedded within them, effectively hardwiring the processes while enabling business rules.

From a market development point of view, the market for standalone BRE tools has failed to materialize. It remains about a fifth of the size of the BPM market. Indeed, some suggest that the recent growth in the BRE domain is entirely due to its use within BPM deployments.

Loose Coupling Business Rules & BPM Engine

To provide greater sophistication, many BPM vendors have adopted what is often described as a “loose coupling” approach, integrating third party BRE tool sets alongside their own BPMS functionality. This allows customers to mix and match, potentially re-using any existing investment in BRE functionality, while enabling the simplification of business processes and externalization of the rules that affect them. This approach also allows developers who are expert in an existing rules environment to continue within a familiar system. However, this approach also delivers a number of disadvantages.

This loosely coupled model, with two separate systems, implies the need to synchronize two distinct object models (inherited class hierarchies) and data models (sets of variables)—one for the BPMS and the other for the BRE. While the underlying objects or data itself is likely to persist in an RDBMS or other application data store, the semantics of how this information is utilized is reflected inside the data models of the BRE and BPMS. Over time, as business needs evolve, changes on one side will inevitably require adjustment of both the object model and data model on the other. The net result? A disproportionate level of effort is required to keep these two models in sync. And when one considers the complexity of the business scenarios to which these systems are applied, that can be difficult. Furthermore, the organization must manage the two different platforms and train staff on their respective development environments.

While the promise of business flexibility is glimpsed, software design decisions force a somewhat rigid solution that can inhibit change and adaptation over time. When using Web Service or API based integration between the BPMS and the BRE, developers need to know what objects are to be passed across—effectively breaking the premise of loose coupling (i.e. that the calling application does not need to know the detail of the invoked application).

Scalability is also an issue. As the process engine of the BPMS calls the BRE, it invokes the identified rule set, passes the appropriate variable data to obtain the decision before continuing on to the next step. This is a relatively “chatty” conversation between the two engines (usually involving the translation to XML, invoking Web Services, etc.) and does not scale well especially in high-performance applications.

Whilst many of the other areas identified in the business opportunity space for rules and processes are possible, they require considerable care in implementation and tend to be implemented by visionary end-users (rather than productized by vendors). Developers are still left with considerable complexity—as decisions get more complex, there are more possible outcomes. Developers must either build more complex rules, or simplify the rules, making processes more complex.

Of course, while it might allow the vendor to put a tick in the box of business rules support and it covers a gap in the reference architecture, this does not necessarily provide the optimal solution to the over-riding business objective.

Business Rules alongside a BPM Engine

Other vendors have developed additional rules-oriented capabilities alongside their process engine. In such situations, the business rules functionality appears as an additional functional area to the BPM environment. More often than not, vendors have stuck to the core notion of supporting sophisticated decision making within processes. A few have implemented better support for resolving the best employee for a given piece of work. One or two have even developed functionality to enable process discovery and ongoing optimization through exception management. A couple of others have used the combination to support dynamically synchronous user interaction (using processes and rules to generate and then drive the user interface). However, most have ignored (or failed to grasp) the wider set of benefits that are possible.

On the other hand, this integrated approach does a better job of resolving the problem of two distinct data models—using integrated “equation editors,” where business rules and processes are developed based on the shared set of variables. But this does nothing to resolve the issue of having two entirely separate object models. Shared data is limited to the variables of the process and its application domain. It stops well short of the sophisticated multi-dimensional object model that is required to deliver the broader functionality described earlier.

BPM Engine, Business Rules and an Extensible Object Model

The highest level of evolution in the BPM arena is where vendors have combined the power of the process engine with business rules and an extensible object model. This approach focuses on providing the sophisticated capabilities of a BRE to support and drive the entire BPM environment; including all aspects of human-to-human, system-to-system and human-to-system processes (rather than merely making better decision). Rules and processes are stored in the same repository, share a common object model, and are designed to work intimately together in all facets of the environment.

The fact that they share a common, multi-dimensional object model ensures ongoing consistency between processes and rules that also allows for constant flux and change. When the rule and process engines share a common object model within a unified architecture, it enables inheritance and specialization of existing process and business logic. This also provides greater flexibility when designing and deploying a cross-functional composite application that delivers BPM benefits to disparate users and groups.

The notion of an integrated object model goes a long way further than the shared process data variables referred to earlier. It allows a particular aspect or context to be specialized down to a particular response that applies in a unique situation. This specialization can extend right down to the level of a single individual or customer (something that is unthinkable using nested IF-THEN-ELSE statements). This is a far more agile and flexible approach (than fixed and pre-determined object models) based on componentized services that can more readily adapt to the rate of business change.

Using this combination of declarative rules, procedural flow and an extensible object model, developers can quickly iterate as they develop and deploy applications. Moreover, the combination enables a wide range of new functionality including:

- Context-Specific Components—processes and rules are dynamically assembled at runtime, based on the context of the case of work in hand. System building becomes an exercise in defining the different contexts and responses to them as integrated components that are bought together as needed (rather than assuming that an individual process will handle all potential situations). Components that are fully encapsulated services that deliver some sort of business capability that, potentially, could apply to individual actors within a specific business context.

- Delegated Development—application development is controlled and delegated to target users, revealing both rules and processes at the same time (rather than separately through different user interfaces). This allows a far more collaborative approach to development, where business and information technology (IT) staff work together to build and modify solutions.
- Dynamic User Interface—screens are dynamically constructed for each individual user (or customer, partner, etc.) at runtime, based on the role they are working in, the requirements of the case in hand, the security model, etc. By selectively granting access to individual applications, processes and rules, the organization can easily implement new business processes or practices to a specific subset of users. At a later time, it can grant access to a wider community, as business needs dictate. The business can also use this sort of functionality to deliver different capabilities to different categories of employees, customers and partners.
- Integrated Event Management—declarative expressions based rules can automatically recalculate values of computed properties, independent of procedural steps. For instance, let us assume that the discount rate offered to a customer is a computed value based on the size of the order, their current outstanding balance, the year to date orders, the status of the customer, number of years, and any incentives offered. If any of these values change, say they paid their outstanding bill (external to the sales order process), then the discount rate will change automatically, regardless of the state of the related process flow. In a normal BPM system, the event would have to be monitored separately and the process would need to invoke the recalculation of the discount rate. This enhances scalability as change-aware declarative rules are used only when underlying data actually changes.
- Backward Chaining—using rules to dynamically work out how to retrieve missing data needed to complete a task (from an external system or another user).
- Integration—defining dynamic interfaces between systems and applications, more directly supporting loose coupling required by Service Oriented Architectures (SOAs). The Service Level Agreements (SLAs) and various complex rules involving decision trees, constraints, when conditions, decision maps, are defined in the integrated development environment. Business people can then use these SLAs to control service operations.
- Extensibility—using the object oriented capabilities of the environment, it becomes possible to create new rule types, enabling developers to expose both the logic and parameters of some desired specialist processing (instead of burying it in custom code), and thus applying the power of rule resolution to otherwise-unique business needs.

Conclusion

Businesses are now better understanding the role and value of automated business processes in organizing their operations. Some have approached it from the other direction, developing sophisticated sets of rules to define and distribute business policies. But few have appreciated how these two approaches should be developed and deployed together; as complementary approaches that really drive business optimization.

Although many of the objectives discussed above are achievable without the use of a unified business rules and BPM environment, inevitably they will require greater clarity of thought and more careful design. Indeed, developing appropriate systems without an integrated object model will only increase the skill levels required. In such situations, highly skilled and experienced people are needed to ensure process and rules are designed and deployed effectively—skills that are often not readily available on the street.

There is a better way, and the industry is evolving toward it now. With an integrated environment—one that effectively fuses business rules, business processes and an extensible object model to form an integrated whole—firms can take optimization to the next level. This new environment is the most effective form of BPM Suite available the whole is greater than the sum of the parts. It delivers a range of new capabilities that are simply not possible with a loosely coupled approach. To fully enable change and innovation that environment must provide a sophisticated specialization capability that can handle multi-dimensional versioning, resolving the correct version of rules and processes based on the context of the situation.

In the table on the next page, the light area represents the amount of work required to implement functionality compared with the level of the “gold standard.” Looking at it another way, the amount of black in the circle indicates the “out-of-the-box” functionality delivered.

Aspect	BPMS	BRE	Loosely Coupled BPMS+BRE	Extended BPMS [1]	Rules Driven BPMS
Process Simplification	◔	◔	●	●	●
Speed of Response	◑	◔	◕	◕	●
Analytics	◑	○	◑	◕	●
Rapid Development	◑	◑	◑	◕	●
Controlled Evolution	◑	◑	◕	◕	●
Granularity	◔	◔	◔	◔	●
Proactive Customer Engagement	◔	◕	◕	◕	●
Exception Handling	◕	◕	◕	◕	●
Multi-Channel Relationships	◑	◔	◕	◕	◕
Compliance [2]	◑	◑	◕	◕	◕
Context-Specific Components	◔	◔	◑	◑	●
Delegated Development	◔	◑	◑	◕	●
Dynamic User Interface	◔	◔	◔	◕	●
Integrated Event Management	◔	◑	◕	◕	●
Backward Chaining	○	◕	◕	◕	●
Dynamic Integration	◔	◑	◑	◑	●
Extensibility	○	◑	◑	◑	◕
Representative Products	Fuego FileNet	ILOG Fair Isaac Corticon	FileNet-ILOG Fuego-Corticon	Chordiant Ultimus Appian	Pegasystems [3]

[1] Different vendors have used their integrated rules features to achieve different goals. The scores reflected here reflect an average with some vendors using the rules features to support a wider range of functionality.

[2] In different ways, all products require some work to implement comprehensive compliance capabilities.

[3] Pegasystems is the only product supporting situational rule and process resolution (dynamic binding based on context) alongside an extensible object model.