

# *Enterprise Need*

*Consolidated & automated decisioning*



## Consolidated & automated decisioning

Rationalization, standardization and service enablement of existing applications will help organizations in putting together an “Enterprise Shared Platform” that can be leveraged in building new applications. Organizations can reduce the cost and time involved in new application development, and application modifications, by reusing existing application components and assembling them into new applications that meet users’ new functionality needs. Also, maintenance budgets can be reduced as a large portion of the organization’s application functionality is delivered by the “Enterprise Shared Platform” thereby allowing organizations to eliminate specialist teams for maintaining each of the disparate / siloed applications.

Externalizing process definitions from applications and hosting them in a specialized system (Business Process Management software) will make them more accessible by all applications thereby making it possible to achieve true end-to-end process automation within a functional area as well as across functions. Also, externalizing the decisions that govern the processes into a specialized system (Business Rules Engines) will help organizations implement dynamic processes that are context/situation sensitive i.e. respond to each situation appropriately. Process automation and dynamic processes help organization achieve straight through processing and respond to Customers faster, thereby eliminating delays and significantly boosting efficiencies.

While the above two initiatives help organization reduce cost, respond faster to changes and gain efficiencies, organizations still need to think of ways in which they can become more effective. Being effective and accurate in determining which Customer to do business with, what price to charge for a product, what level of service to give a Customer based on his/her long-term value, which insurance claim to investigate more closely, which Customer to give a loan to, which market segment to focus sales efforts in, which risk to select by assessing whether it is worth taking can have a significant impact on the bottom line. These high-impact decisions, which in the past required a good deal of human participation, can now be automated using decision systems thereby allowing a large number of factors to be considered in making the decision.

While data-warehousing and analytics software help organizations analyze their past data and market trends to arrive at key conclusions / strategies, organizations need to be able to apply this knowledge to each and every transaction they are engaging in and to each

and every Customer they are interacting with. To translate an organization's knowledge into a software system that can apply the knowledge, in real-time at the time of a transaction, Organizations can implement decision automation systems that leverage technologies such as Business Rules Engines, Product Configuration engines, Price Engines, Constraint solvers etc.

Once such decision automation systems are implemented, when a Customer is on-line or on-phone, Agents and Underwriters don't need to look at manuals or catalogs to determine whether the person is eligible for a product or not, and what the price should be or what product needs to be offered. Such decisions can be taken, automatically, by decision systems such as an underwriting system, product configuration and pricing system etc.

Enterprise Technology initiatives and their benefits to business:

Initiative --->	Rationalization, Standardization and service enablement of existing applications and establishing an Enterprise Shared Platform	Externalizing process definitions, implementing dynamic processes and achieving straight through processing	Data consolidation and analysis to determine key factors, trends & strategies and gaining knowledge	Decision automation and consolidation to automate application of knowledge at the time of each transaction
Benefits --->	<ul style="list-style-type: none"> <li>• Reuse of existing applications</li> <li>• Ability to change applications quickly</li> <li>• Ability to assemble together new applications quickly</li> <li>• Reduce maintenance budgets by eliminating specialists for each application</li> <li>• Quick app integration</li> </ul>	<ul style="list-style-type: none"> <li>• Process definitions easily accessible by all applications</li> <li>• True end-to-end process automation both within functions and across functions</li> <li>• Ability to quickly change processes</li> <li>• Ability to respond quickly and appropriately to each situation</li> </ul>	<ul style="list-style-type: none"> <li>• Knowledge building and distribution to all participants</li> </ul>	<ul style="list-style-type: none"> <li>• Automate complex decisions which, in the past, required a high degree of human participation</li> <li>• Apply the knowledge gained through data analysis to each &amp; every transaction, order, claim etc</li> <li>• Bring uniformity to decision making process</li> </ul>
Business impact --->	<ul style="list-style-type: none"> <li>• Agility</li> <li>• Cost reduction</li> </ul>	<ul style="list-style-type: none"> <li>• Process agility</li> <li>• Improved Efficiencies thanks to straight-through-processing</li> <li>• Higher ROA (return on assets)</li> </ul>	<ul style="list-style-type: none"> <li>• Knowledge empowered workforce</li> </ul>	<ul style="list-style-type: none"> <li>• Right decisions every time &amp; situation</li> <li>• Highly Effective decision making</li> <li>• Higher ROI (return on investment)</li> </ul>

Realizing the benefits of automation and consolidation of "decision making", many Enterprises have attempted to build decision systems such as underwriting, claims adjudication, pricing, product configuration etc. Many of these attempts had a limited success as translating an organization's knowledge into a software system that can apply the knowledge in making high-impact decisions is difficult. Most programming environments do not offer easy ways of representing complex decision logic.

Popular programming languages such as Java are procedural in nature. In these languages, in order to represent business logic (i.e. “what” or facts or knowledge), you have to detail the series of steps that the system needs to execute for implementing the business logic (i.e. “How to” implement the logic). E.g.: *put* some data in a variable; *fetch* the data back out of the variable; *loop* through a block of instructions *until* some condition is satisfied etc. Values are passed from one expression to another by reference to common variables. Further, one expression may change a variable's value before that variable is used in the next expression. Because relationships in procedural languages are expressed using sequences of expressions the order of execution is crucial. If the order in which expressions are executed is changed then the behavior of the entire program will be affected.

To understand this, take a scenario where you are dining at a restaurant. You will tell the waiter what you want to have and how you like it. But, if the restaurant is “procedural”, then you will also have to provide step by step directions on how to prepare the item. Imagine another scenario where in you are visiting New York City for the first time and hired a Tour Guide to show you around. You will let the guide know the various places you want to see (“What”) but won’t give him detailed instructions on how to get to each place (“How to”) as you assume that he is familiar enough with the City to figure that out. But if the Tour Guide were procedural (i.e. not familiar with the City), then you will have to provide detailed directions by looking at a map.

Further, suppose you are visiting only for a day. You will instruct the Tour Guide on what all you desire to see and ask him to show as many of the places as possibly can be visited in a day. As he is familiar with the places and knows time normally spent at each place and time taken to get to each place, he decides the order in which places need to be visited and the route to take. As you are not familiar with the City, it is very difficult for you to provide these details. But if the Tour Guide were procedural (i.e. not familiar with the City), then you will have to figure out these details and instruct the Tour Guide, leading most likely to an suboptimal utilization of your day.

The adjoining diagram depicts how the majority of a procedural program is made up of the “how to” code rather than the “what is” code. Automating even simple decisions (using procedural languages such as Java) requires a tremendous amount of code development.

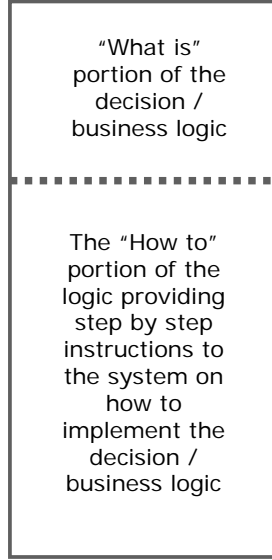
Decision management in Enterprises is complex. Many facts (knowledge) have to be considered in making each decision – whether it be underwriting, product configuration, pricing/rating or loan approval. Also, many a time facts need to be derived from other facts without having to be explicitly stated. If procedural programs have to be written to represent thousands of such interrelated facts, organizations will end up spending considerable resources in developing millions and millions of lines of code.

This is where specialized infrastructure can be utilized. Several vendors have developed robust decision engines that are declarative (not procedural) in nature. When using these engines for decision automation, organizations have to just specify the “what is” portion of the business logic (i.e. they just have to specify the facts) and let the decision engine figure out “how to” take decisions keeping those facts (and derived facts) into consideration. That is, the decision engines execute things that you declare as they know “how to”. Importantly, the order in which the facts have been expressed or executed has no bearing on the decision outcome.

The following diagram depicts how organizations can save a tremendous amount of effort and cost by not coding the “how to execute instructions to the system” as that portion of the work is carried out automatically by specialized decision infrastructure. Also, as the order in which facts are stated has no bearing on the decision outcome, organizations can automate even complex decisions and make changes to them rapidly.

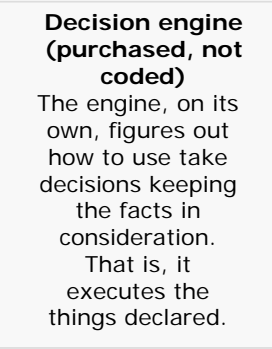
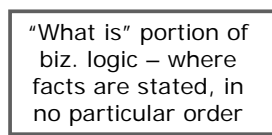
Organizations that have used this technique have realized that, compared to leveraging

**A typical procedural program that needs to be coded**



Operating system, RDBMS, Application Server, etc

**Declarative program that needs to be authored**



Operating system, RDBMS, Application Server, etc

specialized decision engines, developing applications such as underwriting, pricing, product configuration, fraud detection, loan approval, claim adjudications, segmentation and scoring type of applications in procedural environments such as Java etc is simply too expensive, takes a much longer time and very difficult. Moreover, once developed the applications are too difficult to change.

This paradigm of developing business logic is akin to an accountant putting together a financial spreadsheet. The accountant designs the spreadsheet and specifies what formulae to use. But he does not actually provide instructions on how the formulae need to be processed.

In addition to reducing the total amount of code that needs to be developed, Decision Engines provide another tremendous benefit. As the instructions to the engines are declarative and natural language like, business users themselves can participate in authoring the business logic rather than always depending on programmers for coding the logic. Even if some business users or organizations are loath to having business people author the logic, they can at-least make changes to the logic using a workbench like environment and not worry about the change adversely affecting any other logic.

This will cut down the cycle time involved in giving a change request, documenting change request, allocating IT resources to work on change request, making the change, regression testing to ensure that no other programs are affected etc.

Decision Engines have started gaining popularity in the last few years and are expected to become a standard technology that Enterprises will leverage in automating and consolidating decision making activities.

Decisions are made in every function of an Enterprise – be it sales, marketing, underwriting, product management, compliance or finance. Enterprises can automate the decisions made in each of these areas by leveraging Decision Engines. The decision making logic can be organized in such a way that it can be used by applications across the Enterprise.

Some types of decisions involve yes or no answers. Other types involve classifying Customers into segments or risk buckets, generating customer scores, adjudicating

claims, detecting fraud etc. Some other types of decisions involve what product to quote, what price and what discount. Decision logic can also involve development of “optimization models” that can coordinate various decisions being made in the organization, in real-time, and refining them to systematically advance organizational goals. For example, the sales campaign and pricing seem to have worked and orders are flowing in. It may make sense to increase the price.

Applications in a few of these areas are discussed in the following sections.

## Product Life Cycle management, Product configuration

Product Life cycle Management is a decision intensive area. When Product Managers define products, they are essentially specifying decision logic such as what products to be sold for what needs, who is eligible for purchasing the product, what options are available with the product, which channel partners can sell the product, is the product available in certain locations or not, at what price can the product be sold etc.

Once products are defined, the corresponding logic needs to be available to various systems in the company including sales front-ends, billing systems, policy / contract processing systems etc.

The following table provides various functionalities that Enterprises should look for in a Product Life Cycle Management system.

<p><b>An easy to use, GUI driven, environment for Business Users to define products</b></p> <p>Enable Users to quickly define market-aligned products (User Workbench)</p>	<ul style="list-style-type: none"> <li>• GUI driven definition and management of reusable product parts and end products (that are designed for different market needs)</li> <li>• Edit (Modify) existing product definitions quickly in response to competitor strategies</li> <li>• Differentiate product features by distribution channels and private label partners. Private label partners should also be able to make modifications to a product's definition</li> <li>• Copy existing product definitions and make modifications so as to create new products rapidly</li> <li>• Package product parts that are typically sold together and manage them as a logical group</li> <li>• Leverage templates for quickly authoring product rules (that specify how product parts work together to lead to an end product).</li> <li>• Use rule expression techniques that are very intuitive to different stakeholders in product development i.e. Actuarial, Marketing, Jurisdictional Compliance etc. This will ensure rapid and error free product development</li> <li>• Group related product rules into logical rule groups and reuse</li> </ul>
--	--

	<p>these rule groups for efficient maintenance</p>
<p><b>Consolidated repository of definitions and decision logic</b></p> <p>Make product data available to different stakeholders in the organization</p>	<ul style="list-style-type: none"> <li>• Enable rapid product rollouts across the organization by consolidating product definitions</li> <li>• Deliver consistent, accurate product advice, in an automated manner across the organization</li> <li>• Provide intelligent Product Catalogs</li> <li>• Upload product definition updates centrally in one place and make it available rapidly to anyone who requires it</li> <li>• Different stakeholders and applications in the organization can access the appropriate product data (that is relevant to their functions)</li> <li>• Access to Product Calculations, Servicing Rules (for agreements based on a product), Product Documentation, Sales Collateral</li> </ul>
<p><b>Decision engine</b></p> <p>Take automated decisions, in real-time, at the time of a transaction</p> <p>Provide Expert "Product related Advice" to different stakeholders</p>	<ul style="list-style-type: none"> <li>• Based on the transaction data such as needs, location, age, channel, segment etc, recommend the best-fit product for the Customer</li> <li>• Avoid situations where-in the Product Managers have created 100s of products but the Agents are selling only three or four products that are popular or they are most familiar with</li> <li>• Provide the ability to capture information in a structured step-by-step fashion thereby providing the salesperson / CSR / Agent with intelligent advice to understand the Customer, his/her Preferences &amp; eligibility, and offer a best-fit product</li> <li>• Customers can be suggested products in the following ways: <ul style="list-style-type: none"> <li>o Intelligent catalog: a product can be selected from a pre-defined list of product packages</li> <li>o Best-fit configuration: All aspects of the Customer are analyzed in recommending a configuration that is best-fit</li> </ul> </li> <li>• As the engine evaluates all aspects and rules in determining the product to be offered, sales teams need not waste time looking up and interpreting bulky product manuals</li> </ul>



## Underwriting, Risk assessment, selection and classification

Underwriting is a decision intensive area. When Actuaries or Underwriters define risk assessment, selection and classification criteria, they are essentially specifying decision logic such as what is the risk posed by a Customer, what is the expected loss from a particular Customer, how probable is the default risk, which risk class or tier does the Customer fall under etc.

If these criteria are made available as a system, instead of being made available as underwriting manuals, Organizations will be able to take automated decisions and respond quickly to requests from Agents, Private label partners and Customers.

The following table provides various functionalities that Enterprises should look for in an automated Underwriting system.

<p><b>An easy to use, GUI driven, environment for Business Users to author risk decision logic</b></p> <p>Enable Users to quickly define risk causing Factors, their groupings (into Elements) and Weights</p> <p>Enable Users to quickly define eligibility criteria</p>	<ul style="list-style-type: none"> <li>• Risk (of loss or default) can be measured by looking at various factors and attributes. For example, driving violations (Factor) can be an important indicator of the risk posed by a driver. The higher the number of driving violations, the riskier the driver is. Users should be able to quickly define these factors and relations, using a GUI driven workbench, and be able to modify the definitions over a period of time</li> <li>• Once all Factors are defined, using the workbench, Users should be able to assign weights to various Factor to reflect its relative strength in predicting the probability &amp; magnitude of risk (either through single variable or multi variable scoring)</li> <li>• Users should be able to group the Factors into logical groups (Elements) using the workbench. For example, all Factors related to a building such as type of construction, age of building etc can be grouped under the element "Building"</li> <li>• Users should be able to provide weights at the Element level too</li> <li>• Users should be able to define eligibility / knock-out criteria against various Factor values</li> <li>• Users should be able to define various risk classes, tiers or</li> </ul>
---	--

<p>Enable Users to define Classes / Tiers, Segments and Models</p>	<p>segments based on varying combinations of the values of Elements</p> <ul style="list-style-type: none"> <li>• Users should be able to define rules that take into consideration aspects such as Channel, Private label partners, Life-time value of Customer etc</li> <li>• Users should be able to copy existing definitions and make modifications so as to introduce a new underwriting strategy or a strategy for a new market</li> </ul>
<p><b>Consolidated repository of definitions and risk decision logic</b></p> <p>Make eligibility, underwriting, tiering, classification etc rules available to different stakeholders in the organization</p>	<ul style="list-style-type: none"> <li>• Enable rapid rollout of eligibility, underwriting, tiering and classification logic across the organization by making the rules available as a callable service</li> <li>• Deliver consistent, accurate decisions, in an automated manner across the organization</li> <li>• Provide intuitive, easy to understand, explanations for the decision logic</li> <li>• Upload decision logic modifications and updates centrally in one place and make it available rapidly to anyone who requires it</li> <li>• Different stakeholders and applications in the organization can access the appropriate decision logic (that is relevant to their functions)</li> </ul>
<p><b>Decision engine</b></p> <p>Take automated decisions like tiering, risk classification etc, in real-time, at the time of a transaction</p>	<ul style="list-style-type: none"> <li>• Based on the transaction data such as needs, location, age, channel and other factors determine eligibility</li> <li>• Automatically determine the segment, tier or class the Customer falls under</li> <li>• Make the output results available to product engine for product eligibility check, pricing engine for risk based pricing, policy system for appropriately wording inclusions / exclusions etc</li> <li>• As decisions are made automatically, there will be consistent but context sensitive application of the underwriting logic across Customers and transactions</li> <li>• As the engine evaluates all aspects and rules in making the</li> </ul>

	decisions, agents, sales teams and underwriters need not waste time looking up and interpreting bulky underwriting manuals
--	--

## Pricing / rating

Pricing is one of the most critical functions of any organization. Pricing too high can negatively affect revenues and pricing too low can bring down profitability, yet organizations cannot afford to keep a Customer waiting for getting a quote. Many organizations have tried automating this task but ended up with systems either too rudimentary or too sophisticated but rigid.

Organizations can implement automated pricing systems that are sophisticated, flexible and business user modifiable by leveraging calculation engines. Leveraging such systems, Organizations will be able to automatically quote the right price to every Customer and in every situation.

The following table provides various functionalities that Enterprises should look for in an automated pricing / rating system.

<p><b>An easy to use, GUI driven, environment for Business Users to author pricing / rating and fee calculation logic based on risk, product, channel etc</b></p>	<ul style="list-style-type: none"> <li>• Users should be able to quickly author pricing logic and rules</li> <li>• They should be able to load market data and base price data into the system</li> <li>• They should be able to edit (Modify, Copy) different pricing related logic quickly in response to competitor strategies</li> <li>• They should be able to maintain differentiated pricing based on Private Label Client, Channel, Jurisdiction</li> <li>• Different stakeholders should be able to make adjustments at different levels of the pricing logic</li> <li>• Pricing logic should be definable leveraging parameters such as configured products and options, underwriting score, risk class, risk tier, segment etc</li> <li>• Users should be able to copy existing definitions and make modifications so as to introduce a new underwriting strategy or a strategy for a new market</li> </ul>
<p><b>Consolidated</b></p>	<ul style="list-style-type: none"> <li>• Enable rapid rollout of pricing, rate and fee calculation logic across</li> </ul>

<p><b>repository of calculation logic and tables that can be used various applications and front-ends</b></p>	<p>the organization by making the rules, tables, formulae and data available through a callable service</p> <ul style="list-style-type: none"> <li>• Deliver consistent, accurate rates and quotes, in an automated manner across the organization</li> <li>• Provide intuitive, easy to understand, explanations for the discounts, fees and surcharges</li> <li>• Upload calculation logic modifications and updates centrally in one place and make it available rapidly to anyone who requires it</li> <li>• Different stakeholders and applications in the organization can access the appropriate calculation logic (that is relevant to their functions)</li> </ul>
<p><b>Decision / calculation engine</b></p>	<ul style="list-style-type: none"> <li>• Based on the transaction data such as needs, location, age, channel, configured products and options, underwriting score, risk tier, risk class, segment, life-time-value and other factors determine appropriate price / rate and fee</li> <li>• Make the output results available to quoting front-ends, sales front-ends, CRM applications, contract/policy system, billing system etc</li> <li>• As prices / rates and fees are calculated automatically, there will be consistent but context sensitive application of the pricing logic across Customers and transactions</li> <li>• As the engine evaluates all aspects and rules in calculating the appropriate price / rate, agents, sales teams and underwriters need not waste time looking up and interpreting huge price tables</li> </ul>

This white paper is part of a series of papers addressing the topic “Enterprise Needs – 2006-2010. To view other papers, please visit [www.valuemomentum.com/WhitePapers/index.htm](http://www.valuemomentum.com/WhitePapers/index.htm)

### **About ValueMomentum**

ValueMomentum is a global professional services firm assisting Insurance Carriers and Lending Firms by leveraging industry process experience with strong technology expertise, to achieve full potential through the implementation of Agile, Real-Time & Smart Value Chains.

Headquartered in South Plainfield, NJ, ValueMomentum has world class delivery centers in South Plainfield, NJ and Hyderabad, India. There are additional Client Services and Business Development offices in Chicago, IL, Sunnyvale, CA and Hyderabad, India.

#### **Headquarters**

3001 Hadley Road; Unit 8  
South Plainfield, NJ 07080  
Tel: 908-755-0048

#### **ENTERPRISE NEEDS 2006-2010**

1607 N. Aurora Road, Ste 204, Naperville, IL 60563  
Tel: 630-983-6130

20200 Lucille Ave, Ste 12, Cupertino, CA 95014  
Tel: 408-865-0275

#### **Asia Pacific:**

Plot No 1, Sagar Society, Road No. 2,  
Banjara Hills  
Hyderabad- 500034, India  
Tel: +91 40 23556289