

From Business Rules Engine to Business Rules Management System: ABLE Can Handle It All

Abstract

Business rules engines and business rules management systems are complex and poorly understood pieces of software. Exaggerations and misinformation obscure the real capabilities of products as inadequately maintained and improperly documented software masquerade as ‘ideal candidates’ for organizations. The key to choosing the correct business rules management system is understanding the terminology and understanding what a business rules engine and business rules management system should do. A poor choice in business rules management solutions can result in the failure of a project. Business rules engines and business rules management software should be simple to implement, simple to use, and simple to write rules for. Rules should be able to be written by business users in natural language, or by IT. ABLE, Mythic Development’s core business rules engine is available as either the ABLE Depot, a light-weight BRMS, or as the ABLE Business Rules Engine. This paper will examine some of the pitfalls of business rules engines and business rules management systems and explore possible solutions to common industry problems.

Contents

I	Introduction	1
II	Background	1
	The Market Today.....	1
	The Largest Hurdle	1
	BRE vs BRMS	1
	DIY Business Rules Engines	2
	Open Source BRMSs	2
	The Interface	2
	Bridging Business & IT	2
	Handling Complex Rules.....	2
	Potential Benefits	3
	Types of BREs and BRMSs.....	3
	Current Market Solutions.....	4
III	A Real Solution	4
	ABLE	4
	Included in ABLE:	4
	What Are the Benefits of Using ABLE?	5
	Why is this important?	5
	Examples of ABLE’s Benefits	6
IV	Conclusion	8

I Introduction

Business rules engines (BREs) and business rules management systems (BRMSs) are complex. It is difficult even for industry experts to choose the best fit for their organizational needs. There are many different business rules management systems on the market and choosing the right one is vital. Choosing a poorly constructed or poorly maintained BRMS can cripple a project causing a drain on time and resources and ultimately dooming a project before it can get off the ground. The cornerstone of a BRMS should be simplicity. ABLE business rules engine is the central pillar of Mythic Developments BRMS, called ABLE Depot. Simple rules, implementation and ease of use are key factors in the ABLE BRMS. This paper explores business rules engines and business rules management systems and their implementation of these key factors.

II Background

The Market Today

The current market contains a variety of different options for BREs and BRMSs. From open source to multi-million-dollar yearly contracts, it can be unclear which option will save an organization the most time and money and be the best fit. Some companies that once offered BREs and BRMSs no longer maintain their products. This results in high levels of frustration for organizations that chose the wrong product and can result in projects collapsing due to elevated costs and an inability to use the purchased system.

Another common issue with BREs and BRMSs is that they are not language agnostic and lock organizations into a specific language. Proprietary engines can also necessitate additional software to interface successfully with existing systems. Some BRMSs require up to a dozen or more products and services be purchased, with individual support packages for each product and service. Additionally, the different types of Business Rules Engines, Business *Rules* Management Systems and Business *Process* Management Systems makes choosing the best fitting solution next to impossible.

The Largest Hurdle

The hardest part to choosing a BRE or BRMS is knowing the capabilities of the systems. Each system's capabilities are so drastically obscured or proprietary that understanding them – which is critical to deciding which to buy – is a lesson in complexity. Every BRE and BRMS software system is so unique that comparing them is almost pointless.

False promises and murky terminology make evaluating a solution difficult at best and nearly impossible at worst. Low up-front costs do not always translate into low costs down the road, and investing a substantial amount of money up front does not necessarily mean a streamlined development process as proprietary BREs and BRMSs typically require multiple layers to access them.

Additionally, some BREs work well with only a small number of rules, but fail with complex nested rules, while others handle nested rules capably but do not provided the speed required. Some BREs require rules that are written in complex code, some require proprietary syntax, and some will not work unless compiled into the organization's source code.

BRE vs BRMS

A business rules engine handles processing rules, while a business rules management system provides an interface to handle inputting rules, verifying rules, and *should* bridge the language gap between business and IT. The needs of an organization – and who will be handling the business rules – defines whether a full business rules management system is needed. In smaller organizations, a full scale BRMS can be costly overkill, and in large organizations only having a BRE is not enough to bridge the language gap.

Implemented by developers, a BRE works behind the scenes to process business rules. It needs to be fast, capable of complex calculations and comprehensive enough to work throughout an organization's infrastructure. Conversely, a BRMS is implemented by the vendor as a user-friendly way to interact with a BRE. Simplicity, flexibility and portability are the cornerstones of good BRMSs.

DIY Business Rules Engines

When faced with the need of a tool as complex as a BRMS, a lack of understanding of the options leads many organizations to build their own custom BRE. Building a custom BRE is a complex and time-consuming process. A BRE stands at the heart of an organization's software, allowing control of the ebb and flow of productivity.

Homegrown solutions run the risk of explosive growth resulting in cobbled together pieces that don't fit the original architecture and are added in after the fact to suit the needs of a growing organization. For this reason, homegrown BREs tend to be slower and lack the flexibility needed to handle complex rule sets. Rapid growth and implementation can cause these engines to have simplified interfaces unable to sufficiently bridge the gap between business and IT making new rules hard to implement.

Open Source BRMSs

Open Source BRMSs are one option that organizations explore. Open Source BRMSs require modification and support to successfully implement. To alleviate this need, some Open Source BRMSs offer Enterprise options and support for corporate clients.

Open Source BRMSs offer a low upfront cost, but have hidden costs later on. Those hidden costs often tally up to more than purchased systems. Open Source systems are a development commitment for the organization. They are often slower, language specific, more complex to implement, less reusable, unreliable, and not as user friendly as purchased alternatives.

The Interface

A BRMS is an interface for manipulating rules that are processed by a BRE. The design of this interface is absolutely critical. After years with a single BRMS every part of an organization will be familiar with this screen. For this reason, even little quirks need to be considered a severe weakness in a BRMS. Purchased BRMSs typically only provide a single visual interface for controlling business rules as a business user. An effective BRMS must be easy to learn, diverse enough to fit any facet of the organization, and open to modification as an organization grows and adapts.

Bridging Business & IT

Business users often have trouble communicating with IT, creating difficulties translating organizational needs into appropriate software. By providing readable rules that are understandable to both business users and IT, a BRE provides a bridge between the two sets of users. A BRMS takes that capability and broadens it, allowing business users to write and control their own rules without requiring extensive meetings over what a rule should do.

Many BRMSs do not fully bridge this gap and support only a direct interface with the underlying code or use a clunky interface that requires a proprietary syntax to attempt to approximate natural language. Many BRMSs fall short on this goal by specifying a mandatory interface with a mandatory language and syntax. When BREs require awkward formatting – or are not flexible in their specific 'general language requirements' – they can create more headaches than they solve.

While a good BRMS bridges the gap between business users and IT, without a good natural syntax the system damages the organization by increasing animosity between departments. It is critical that a quality BRMS support the languages and syntaxes of the organization implementing it. Any proprietary or incomprehensible tools within a BRMS destroy its usability and turns its benefits into burdens.

Handling Complex Rules

Organizations are complex. A BRE that cannot process complicated rules is *useless*. Handling complex rules means:

- Processing complex rules quickly
- Making them clear to the business users
- Providing comprehensible access for development
- Providing the ability to alter parts of existing rules
- Handling nested/multi-tiered rules
- Handling rules that depend on other rules
- Ensuring order of operations
- Restricting rules to run under organizationally relevant circumstances

Missing any one of these pieces makes a BRMS, and the attached BRE, impractical for any organization with reasonable complexity. There are few, if any, current market BREs that can truthfully claim to fulfill all these requirements.

Potential Benefits

A good BRE separates the ‘knowledge’ from the ‘workflow’. It will not magically simplify the complexity of the organization implementing it, but it will simplify future maintenance and upkeep of a software system. It will not remove the need for oversight, but it assists in overseeing aspects of the organization and reducing the time involved in upkeeping business rules. It will not create its own rules, rather it will provide a simplified and streamlined process for implementing rules.

There are many benefits to picking a well-built BRMS that suits an organization’s needs. Not only are development costs reduced, but continuing maintenance costs are mitigated. Separating software from the rules of the organization allows growth without massive development costs. It also allows the organization’s needs to be managed by subject matter experts.

A BRMS should...

- Provide a quick, easy to use interface
- Make rules easy to understand by providing the ability to write rules in the language of the organization.
- Not be a long, complicated process to implement.
- Provide the ability to write complex business rules to fulfill real world needs.

An effective BRMS should...

- Help bridge the gap between software developers and organizational managers.
- Make complex rules reliable and fast, scaling to the size of the organization.
- Support multiple tiered rules to allow for real world complexity, not just syntactic business rules.
- Ensure that tiered rules output reliable results.

An ideal BRMS should...

- Provide options for interacting with business rules including custom interfaces.
- Make creating complex, multi-tiered rules simple, containing as few steps as possible.
- Support limitless tiers of complex rules
- Process rules nearly instantaneously

Types of BREs and BRMSs

There are many terms associated with BREs and BRMSs. Understanding them allows organizations to better determine their needs.

Engine Term	Example	Meaning
Forward Chaining	$X + Y = Z$ balance + deposit = total	Rules are applied and stacked to get a result. Inputs are processed to provide an output.
Backward Chaining	$Z = X + ?$ $10 = 5 + X$	The result is analyzed to guess which actions caused it. Outputs are processed to infer inputs.

Rules Term	Example	Meaning
Production/Inference	If ____, then ____	Information is taken, rules are judged, and verdicts are returned.
Reaction/Event Condition Action	When ____, then ____	Waits for an event to occur, then reacts.
System Term	Definition	
Business Process Management	Defines and automates business processes to make an organization more efficient.	
Business Rules Management	Manages organizational rules that control organizational decision making. Often refers specifically to software systems.	

BREs are typically forward chaining engines primarily utilizing production/inference rules. Part of the confusion with BREs and BRMSs occur because some Business Process Management Systems claim to be Business Rules Management Systems and many BRMSs attempt handle both business rules and business processes through complex event processing engines which use forward chaining, backward chaining, production and reaction rules. These contradictory formats are never implemented effectively and create additional market confusion, slow the implementation process, and tend to destroy the simplicity that a BRMS is designed to provide.

Current Market Solutions

Learning about an existing BRMS requires a massive investment of time, money, and effort. Finding understandable documentation on existing BRMSs is difficult, and short of implementing each one, there is no way to receive a clear picture of how a BRMS works. The shared terminology above should provide a clearer picture of the way a BRMS will enhance an organization. The challenge lies in the lack of understanding of that terminology which extends even to BRMS producers. Wouldn't it be nice if every BRMS provided an upfront breakdown of the true capabilities and weaknesses of their systems?

Open Source engines are largely written in Java, which makes them slow at processing large numbers of rules. Some come with tons of pieces which are confusing to learn and use. The rules use Java Objects, which prevents business people from entering the rules. The language is difficult for non-programmers to understand and it does not bridge the gap between business and IT successfully. They are usable rules engines, but fall short in crucial ways.

One company has a complex suite of products with multi-million dollar upfront costs. Each product must be licensed separately. To get the full suite of programs, twelve or more programs must be initialized and subscribed to. Due to the complexity of their BRMS, it is suggested that an extra 20% be spent on support for each product. Since this is a proprietary engine, only their own software is built to interface with it, so suddenly there are many layers between the program and the core engine. Wouldn't it be nice to have a fast, easy to use, engine that can be accessed through a simple API?

III A Real Solution

ABLE

The ABLE BRMS, called the ABLE Depot, is an all-inclusive, light-weight business rules management system that uses ABLEngine, a forward-chaining engine utilizing production/inference rules. The ABLE Depot offers a scalable solution that fits the needs of every organization. Whether used as a BRE or a full scale BRMS is needed, ABLE is a light weight system that can fulfill all of an organization's needs. No complex implementation or specialized software is needed. Rules can be written in JSON format, or can be updated via a simple UI, allowing fluid communication between business users and IT.

ABLE Includes:

ABLEngine (BRE):

ABLEngine is the BRE that stands at the heart of the ABLE Depot. ABLEngine is fast and language agnostic, capable of handling complex, multi-tiered, production rules. ABLEngine rules can answer questions or set values, to allow more complex rules to be built and utilized. ABLEngine allows multi-tiered rules, and allows rules to be categorized into groups to ensure that the correct group of relevant rules are run at the appropriate time. ABLEngine takes the complex world of business rules and simplifies them to a syntax that is readable by an average user.

ABLE Station (User Interface):

The ABLE Station allows a simple interface for business users to enter business rules. It also provides basic rules checking, validating and verification. ABLE Station is customizable to an organization's specific needs allowing an organization to make the UI as simple or complex as needed. ABLE Station allows users to speak in the language of their industry while providing the full functionality of a complex business rules engine.

ABLE Relay (REST API):

ABLE Relay provides an interface through a REST API allowing access to developers and organizations with even the most complex customization needs. This means that whether local or remote, app or desktop based, the organization's business rules are always accessible. ABLE Relay can sit behind ABLE Station, giving organizations with global reach access to consistent rules everywhere in a format that every user can understand.

ABLE Platform (Local Access):

The ABLE Platform allows ABLEngine to be run locally to directly interface with software without any network lag. This provides increased security and reliability when needed. The Platform is another convenient method that Mythic Development offers to simplify interfacing with ABLEngine. The unique implementation of ABLE Platform makes it language agnostic so developers can continue to build organizational software in the language of their choice.

ABLE Firers (Testing):

ABLE Firers expands ABLE Station to provide active testing to allow users to tweak new and existing rules without changing live data unnecessarily. This provides an added layer of reliability to ensure that the correct answers are being returned every time.

What Are the Benefits of Using ABLE?

ABLEngine:

ABLEngine provides an increase in speed and security over current market BREs and BRMSs. Its unique method of handling business rules ensures order of operations, restricts rules to run under organizationally relevant circumstances, handles nested rules and handles multi-tiered rules flawlessly. This makes it as robust as a programming language without proprietary syntax. ABLEngine uses a simple ‘if this, then that’ syntax to improve readability of rules.

Another core benefit of ABLEngine is that it is simple to implement. ABLEngine is language agnostic and does not require complex integrations to make it work. Given ABLEngine’s ability to handle rules of any complexity, no additional solutions are required to make ABLEngine work.

ABLE Station:

Simple. Customizable. ABLE Station can be built to fulfill an organization’s needs. A basic interface is provided as part of the ABLE Depot suite, but ABLE Station is fully customizable. Either build it inhouse or contract Mythic Development to build it to specification, this customization option allows business rules to be entered, validated and verified in the most convenient format for the organization. Offered as a service or implemented inside the organization, ABLE Station is flexible enough to fit any need.

ABLE Station also allows rules to be exported as a JSON file, allowing an organization a simple option for backing up business rules. Organizations can choose to export individual rules or the entire ruleset. Exported files can later be imported or reimported as needed.

ABLE Relay:

ABLE Relay benefits organizations by providing a simple interface for one or more programs to touch the same set of core business rules. By using an online API, ABLE Relay can provide business rules to one or more programs or interfaces to meet the needs of a large organization. ABLE Relay provides secure data transfer and reliable results regardless of the complexity of rules being processed.

ABLE Platform:

The ABLE Platform allows direct local access to ABLE’s core engine. This means enhanced speed and security, with no waiting for slow networks, and no additional levels of processing needed. ABLE Platform is an ideal solution for organizations that deal with sensitive and restricted data or need speeds above network bandwidth.

ABLE Firers:

ABLE Firers provides a reliable test platform to make rules more robust and reduce the confusion of what the rules will return. ABLE Firers provides the benefit of being able to see exactly what a rule will do before it is used in production.

Why is this important?

The suite of software included in the ABLE BRMS allows organizations to take complete and active control of their business rules and business rules management software. By using simple program interfaces, Mythic allows organizations the option to either use the full ABLE BRMS suite, or to build and use their own solutions around the core ABLE business engine.

By allowing a customizable business rules entry interface, organizations can choose what works best for them. This can be as simple as a single interface, or as complex as multiple interfaces to support various groups and users. Rules are written in natural language making them simple to understand. This creates flexibility to set values to real-world concepts. For example, an organization in the steel industry might set: “ON HAND INVENTORY = BILLETS + SCRAP METAL – SHIPPED TONNAGE”.

ABLE business rules are as simple as “if this happens, then do that.” ABLE is intended to be utilized inside of other programs, and to provide complex logical analysis distilled into a simple, easy to use, format. ABLE allows an organization to separate the business logic from the business processes, which means that updating rules does not break the core program.

ABLE’s light-weight structure allows an organization to choose how much of the ABLE Depot that the organization wishes to use. By separating out business logic from business processes, ABLE also allows companies the choice to control which processes are tied into the business rules logic. Whether an organization wants to tie in every item, from printers to faxes, or simply wants to control core business functions, Mythic Development believes that is a personal choice to be made by the organization, and not a choice that should be imposed by a BRMS.

Examples of ABLE’s Benefits

ABLE & TAUS

By partnering with Mythic Development, LLC, Loan File Intelligent Review Systems (LFIRS) created the Total Automated Underwriting System (TAUS). TAUS allows users to underwrite a loan application (see what the requirements for that application will be). By utilizing ABLEngine and a comprehensive set of business rules, LFIRS created a specialized interface that makes it simple to underwrite a loan.

LFIRS’s TAUS software can reduce the amount of time needed for an underwriter to process a loan from upwards of four hours down to roughly half an hour. By running every rule every time, TAUS reduces the amount of research required to complete an underwrite *and* it reduces the number of scratch-and-dent loans. This means that the productivity of the underwriters is increased, and the number of errors made are decreased because every rule is run on every loan so that an underwriter’s bad day is not reflected in the quality of the underwrites produced.

TAUS is one example of ABLE Depot in use in the real world making real money for real organizations in a reliable and consistent way. ABLE’s flexibility allowed underwriters to design business rules in natural industry language, referring to LTV, DTI, and other industry terms without complex configuration. The innate ability of ABLE to allow the organization to define their own natural language means these same capabilities are available to any organization in any industry.

ABLE & DART

Actual organizational decisions are simple but wide scale. Small scale concepts build on each other to form complex decision chains. In the end, large decisions is often as simple as yes or no, but they require many small data points to correlate accurately to form a decision. ABLE provides the simple, comprehensive, rules in that natural language of yes or no. DART grants access to enough data for ABLE’s answers to be accurate.

Because of ABLE’s flexibility, each department can have individual sets of business rules that grant a simplified view of that department’s status. This allows executive level business rules to evaluate the whole organization, utilizing the departments’ statuses, to return overarching decisions.

ABLE’s simplified business rules combined with DART’s ability to read data from any source, means that an organization’s data sources can all feed data to overarching organizational business rules. No longer must an organization repeatedly create the same rules for each department, instead ABLE rules apply to every aspect of the business equally. If the organization were dealing in stocks, DART could read data on numerous stocks in a single data set and ABLE could process rules to make decisions on when to buy and sell based on actual data instead of just a guess. This combines the instant access to data of a computer with the complex decision making of an expert stockbroker.

ABLE & Pharmaceutical

ABLE’s ability to not only use natural language but also complex programming allows detailed rules that are far superior to a non-agnostic BRMS. When dealing in pharmaceutical research, this can mean the difference between life and death. The rules can be made in such a way that it watches over a team’s project, freeing up the team to do additional exploratory research into more efficient chemical makeups. ABLE allows scientists to focus on science instead of wasting their time writing programs.

When scientists have assembled a proper procedure, they can enter the rules and allow ABLE to handle the minutia. As anomalies appear, ABLE allows scientists to handle them instead of simply terminating the program or experiment. Creative use of ABLE allows specific flags to be placed, identifying issues that need immediate attention or are less severe.

Additionally, scientists could combine DART with ABLE. DART would allow the unimpeded sharing of data by colleagues or organizations across the globe while ABLE would allow teams to create collaborative rulesets so that the scientists can concentrate on what works. This data and ruleset sharing unifies research and keeps everyone informed as if they were working in the same lab.

ABLE & Automation

ABLE's natural language rules provide the ability to mimic human creativity in complex applications. While consistent, reliable, processing is the cornerstone of ABLE, the natural language input adds human creativity into the processes, providing far greater functionality, reduced downtime, elevated safety, better security, and increased flexibility. In cases where a human is not directly involved with a process, ABLE allows complex rules to ensure automated systems function as expected and can handle unlikely situations.

Today's machine learning algorithms make a lot of mistakes. Over time, with more data, occasionally the machines get it right. With ABLE, machine created rules can be evaluated by humans and input directly into a process to create better systems minimizing the risk of errors.

If an organization were building software for an autonomous machine with limited connectivity, the software needs to be capable of handling complex decision making while waiting on updates. An example of this might be off world mining where complex environmental variables, like those on the surface of an asteroid, require decisions with decades of research behind them. Only a human could effectively make those decisions before ABLE. Whether a mining rover on one of the moons of Jupiter, or exploring the surface of Mars, ABLE creates reliable, consistent, human decision making when direct oversight is not possible.

IV Conclusion

Choosing a BRE or BRMS is complex. Confusing terminology and obfuscated abilities make evaluating BRMSs even more difficult. Mythic Development's ABLE provides an easy to use BRMS and BRE that can handle complex, multi-tiered business rules quickly and in a language agnostic format. ABLE provides multiple methods to interact with the core engine and all interfaces are fully customizable. *This means that regardless of the size of the organization, complexity of rules, or language or languages used, ABLE is a custom fitting solution.* Whether a single business rules user interface is needed, or multiple user interfaces are needed, Mythic allows users to build their own, or can be contracted to build custom interfaces for the ABLE BRMS. ABLE's use of natural language means that specialists in the field can write the business rules, and that business rules are no longer only the domain of the IT department. ABLE is available as either an internal implementation or an external service and contains an engine, a business rules user interface, REST API, local access to ABLEngine, and an active testing platform to ensure that an organization's every need is met.

Schedule a demonstration today.

<https://mythicdevelopment.com/contact>

(321) 900-3876

sales@mythicdevelopment.com