

by David Wright

***“The biggest risk to your company is not being able to change fast enough... Business Rules are the answer.” ...Ron Ross***

I am a great appreciator of Mr. Ross. He has written extensively on the topic of Business Rules, offers excellent training on the subject, and is the keynote speaker at each year's International Business Rules Forum. I would like to start my own article on Business Rules with an 'icebreaker' he used on a seminar I attended.

Consider the sport of American Football. Some aspects of the game are very stable, some less so, and some not necessarily stable at all.

Highly stable aspects are the defined Terms/Concepts shared by all participants, such as:

- team, home, visitor
- yard line, touchdown, tackle
- field, play, down, penalty
- coach, referee

Almost as stable are the following Facts/Relations:

- Visiting team plays home team
- Field is 100 yards long
- Team runs play
- Player makes tackle

and Game Rules:

- A game must be played by exactly two teams.
- Points must be scored in increments in 1, 2, 3, or 6.
- The team with the ball must move the ball 10 yards in four or less downs to receive a first down.

Items that not very stable are the Plays, such as:

- End Around Sweep
- Flying Wedge
- Quick Kick
- Post Pattern

Sports are one of the few enterprises that value the existence and management of a 'rule book', for consistency and to support decisions. Rules do change over time, as the 2 point conversion is a newer addition to the professional game. However, low level procedures, like Plays, can change and new ones be introduced without having to change the Rule Book.

Each enterprise will have its own concepts, facts and rules to guide it; northern North Americans already know that Canada has its own football enterprise, with different rules such as 3 downs and 12 players per side.

So again, review the range of stability. New terms and concepts are introduced much more rarely than actual rules or procedures guided by those rules. Out of this we can see that Rules are based on the following three Components:

- *Terms*
- *Facts*
- *Rules*

These components are alike in the following ways:

- They are shared in an enterprise, nothing is secret or private.
- They are declarative, not procedural.
- They are persistent, not temporary.
- They are structured, not casual.

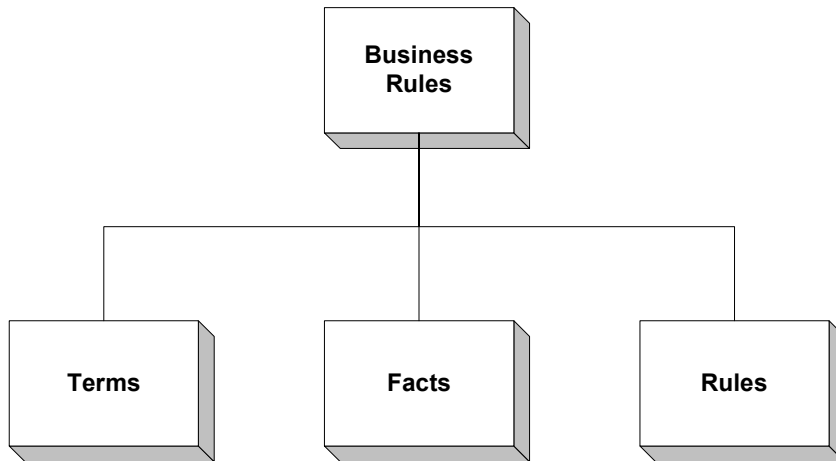
What are Business Rules, really? From a systems perspective, at least, Business Rules are atomic pieces of re-usable business logic, specified declaratively. They cannot be broken down further without losing meaning (atomic), they can apply across many processes and systems (re-usable), they are not programming logic (no IF-THEN-ELSE statements), and they do not state how they are enforced (declarative).

More formally, a business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business. (Business Rules Group, 2000, see <http://www.businessrulesgroup.org> ) A Business Rule Statement should be a natural language sentence, considered a requirement, strictly non-procedural, and an expression of an operational business policy or practice.

“From a *business* perspective, a business rule is a directive that is intended to influence or guide business behavior. Such directives exist in support of business policy, which is formulated in response to risks, threats or opportunities.”

*From Business Rules Group, 1998  
(formerly GUIDE Business Rules Project)*

Business Rules also come in different flavours; to reiterate, a high-level business rule classification system is shown as follows:

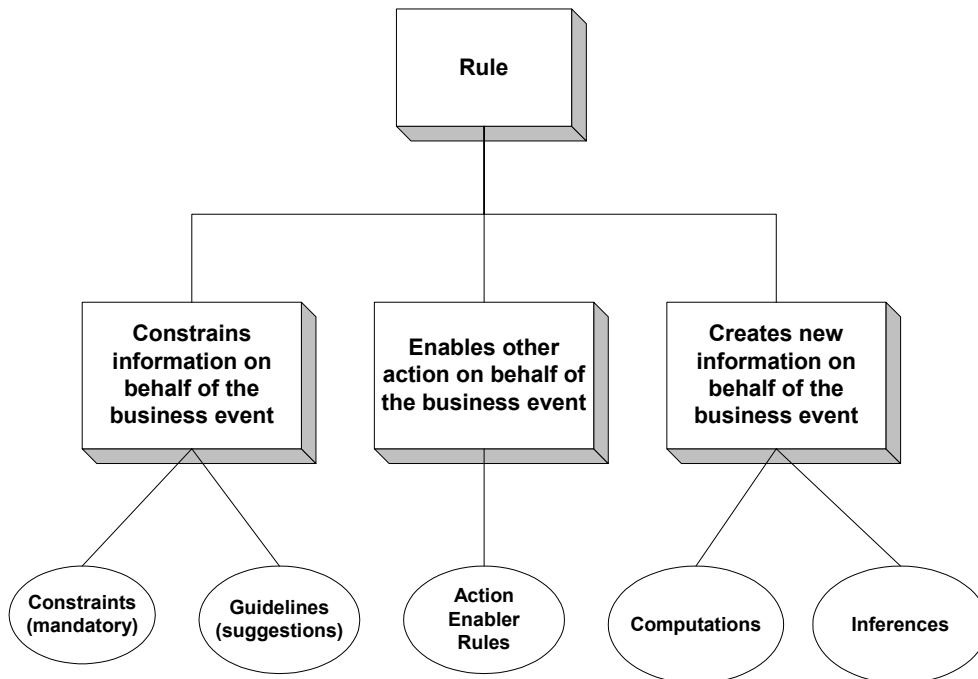


A **Term** is a noun or noun phrase with an agreed definition, usually describing an item of business interest like *Customer* or *Credit Application*.

A **Fact** is a statement that connects Terms, through prepositions and verb phrases, into sensible, business observations; for example, *A Customer submits one or more Requests For Credit*.

A **Rule** is a declarative statement that applies logic or a computation to information values, resulting in the discovery of new information, or in a decision about taking action. For example, *a Customer requesting credit must be at least 18 years old*.

A further classification of Rules is shown below:



A constraint can be a mandatory or suggested restriction on the response to a Business Event. A **Mandatory Constraint Rule** is a complete statement that expresses an unconditional circumstance that must be true or not true for the business process to complete with integrity. For example, *a Customer must not have more than 5 loans at any one time.*

A **Guideline Rule** is a complete statement that expresses a warning about a circumstance that should or should not be true. As a warning, it informs the appropriate person, but allows the person to make the choice. For example, *a customer should not have more than three Requests for Credit active at one time.*

An **Action Enabler Rule** (“Action”, for short) is a complete statement that tests conditions and upon finding them true, initiates another business event or internal activity/process. For example, *a credit check is performed for all co-applicants if the Credit Request is over \$250,000-.*

A **Computation Rule** is a complete statement that provides an algorithm (usually mathematical) for arriving at the value of a term. For example, *the regular payment for a lease is the base lease payment amount plus any applicable provincial and/or federal sales taxes.*

An **Inference Rule** is a complete statement that tests conditions and upon finding them true, establishes the truth of a new fact. For example, *if a customer is a Platinum risk, the interest rate on their new loan will be 1.5% below the current prime rate.*

Why do we need Business Rules? To change our processes and systems as fast as the business changes!

“When the rate of change increases to the point that the time required to assimilate the change exceeds the time in which the change must be manifest, the enterprise is going to find itself in deep yogurt.”

John Zachman

If speed and flexibility are the key to business success today, then any organization needs to change its products and operations in near-real-time. Much of how an organization runs today is dependent on its processes and systems, where their policies, procedures and rules are embedded in computer system code, even if they don't perceive it in these terms. However, once the business realizes they need to change their processes/systems, they often hear from IT that it will take months (and sometimes more months) to make the needed changes. That sounds like a lot when you want to change some simple rules, such as decreasing GST by a percentage or changing the start date for Daylight Saving Time.

How did we get to this point? In the beginning we had data processing, emphasis on the processing. In relative terms, it was not long before the need to separate process and data was recognized, leading to database management systems and techniques like data modeling to support them (all hail Codd!...and Chen and Finkelstein et. al.)

Data Models introduced the first Business Rules, such as optionality and cardinality of relationships, although they were not called rules per se. However,

data models did not and cannot document the kinds of rules described above. Readers of Mr. Ross' past publication, *The Database Newsletter*, may recall his attempts to extend entity-relationship diagrams with additional constructs that would document Rules graphically; they did not work out as they made the diagrams too complex to understand.

On the other hand, leaving Business Rules in the process portion of systems was not acceptable for the reasons already mentioned above. This issue was taken up by GUIDE and its Business Rules Project. This eventually led to the "Business Rules Manifesto" (see <http://www.businessrulesgroup.org/brmanifesto.htm>) and the establishment of the Business Rules Group out of the GUIDE project.

The sub-title of the Business Rules Manifesto is "The Principles of Rule Independence". What this is telling us is that Business Rules are independent of both Process and Data, creating a third major component of system definition. Processes use Rules, and Rules use Data, but separating Business Rules definition and execution from the other two allows the business to manage and change their Rules directly.

This separation also improves the Data and Process components: DBMS' can focus on data storage without need to use stored procedures, while programs can focus on implementing common functions with exception handling be managed by rules<sup>1</sup>.

And, just as factoring out Data as separate systems component gave rise to DBMS', factoring out Business Rules is being supported by tools called Business Rule Engines. While it is possible to develop one's own Rules component of a system, many Rule Engine products are already available from vendors. They normally offer (or bundle with) a Rule Modeling or Documentation tool where Rules are recorded, and an execution engine that runs the Rules as needed. Such engines would be called by a system program when a decision is needed or choice of action to be selected based on the results of a Rule. The Engine accepts data from the program, or gets it directly from a DBMS, and tests the Rule (often called 'rule firing') and returns the result to the calling program. Specifics of the Rule are hidden from the program, and can be changed without impacting the calling system.

---

<sup>1</sup> The Process component of systems is also being improved by the factoring out business process flows, supported by new BPM tools, a subject I want to address in a future article.

Yes, but is it real? Does it work? I would not have written this far if the answer was no, but there are means of finding examples, such as:

### 1) User Success Stories

The Business Rules Community now has its own dedicated conference, the International Business Rules Forum. Some of the companies that have presented their experience with Business Rules are:

The Hartford:

From 2003 to 2005, the Hartford implemented Rules-driven processing for underwriting all new and renewal insurance business. Over 2,000 rules were defined but it was discovered that about 30% of the rules previously implemented in systems could be retired. New rules were added as well, and the underlying system is now handling 100,000 plus transactions a day with sub-second response.

Blue Cross Blue Shield of Minnesota:

This health insurance company first implemented a rules engine in 2002, and has seen significant ROI from the automation of claim adjudication business rules. They admit focusing too much on automation to start, without an overall Governance Structure; they now have such a structure for managing Rules, from definition to implementation, subsequent changes, and eventual retirement.

At this point, they are managing 30 to 40 rule changes per week. Changes are documented, implemented in a test Rule environment and then follow into integration testing etc., until implementation. Their cycle takes about 2.5 weeks from change approval to implementation. Four full-time Rule Analysts manage the whole process; their IT department is only involved once a year to implement the rule engine's latest release from its vendor.

Other success stories I have seen include GE Energy, UICI/HealthMarkets, SunGard, AIG Agency Auto, Fannie Mae, Washington Mutual Bank, Experian, and Logistics Management for the US Department of Defense.

### 2) Vendors configuring their products with Rules.

The problem with Commercial Off The Shelf (COTS) software products over time was that they could not meet the 'unique' requirements of any one customer. Companies had to struggle with choices like "we will change our business to match the package", or modify the package code to meet their requirements at risk of losing vendor support or the option of implementing future releases.

But what are the most likely 'requirements' that are unique to a company? Its' Rules. Vendors have come to this conclusion as well; while they may have started with 'table-driven' systems or speak of 'system configuration", my own recent experience with packages has shown that defining Rules is the key to implementing a package that meets your unique needs without changing code or losing vendor support.

One example is Kronos, a vendor of HR Timekeeping systems. My company needed an application to record attendance, vacation and overtime to replace an excel-based 'system' despised by already busy managers. As part of acquiring a solution, we documented our requirements --- process, data, and rules --- mainly to evaluate products but, after selecting Kronos, the requirements were used as input by the vendor's application analysts to configure their system; they were particularly pleased that we already had defined the necessary rules. Two analysts spent one month on site configuring, we tested for one month, and then went live. No code was touched, and no programmers were part of the project team.

On a larger scale, we are currently evaluating loan/lease origination systems, similar to what your car salesman uses when you finance a new vehicle; one vendor's product (to remain un-named at this point) features user defined workflows and rules. The application is delivered in a 'clean' state or with a 'vanilla' configuration that can be adjusted as needed. At this point we have completed reference checks with existing customers, who have all spoken to implementation based on configuration of processes, user-defined data fields and rules. Subsequent 'maintenance' is based on changing rules and adjusting workflows, performed by business experts rather than IT staff. Of course, we are only hearing from satisfied customers, but the message has been consistent so far.

So, I finish this article with a plan to return down the road to report on how this project makes out. I believe that rules will play a key part in its success. In the meantime, I plan to start a SIG on Business Rules on this site and encourage everyone's participation.