KnowGravity Inc.
Badenerstrasse 808
CH-8048 Zürich
Tel: +41 (0) 1 434 20 00
Fax: +41 (0) 1 434 20 09
www.knowgravity.com

Rules that Rule the Business

The Business Rules Approach - Concepts

26. July 2002

DOCUMENT HISTORY

Date	Author	Modification	Chapter
7.2.02	M. Schacher	Initial version	(all)
26.7.02	M. Schacher	Minor changes after review	(several)

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

1.1 About this Document

This document is part of the series "Rules that Rule the Business" about the "Business Rules Approach" (BRA). The goal of this approach is to focus the IT solutions within an enterprise holistic on its business requirements. This document explains the underlying concepts of the Business Rules Approach as well as its chances and risks when it is introduced as a paradigm for developing business oriented IT systems. The other documents within the series discuss the following topics:

- [KG02a] illustrates the Business Rules Approach based on a simple case study by illustrating the main differences between conventional IT systems and IT systems developed using the Business Rules Approach.
- [KG02b] discusses the consequences of the Business Rules Approaches on business as well as software engineering and outlines a development process for Business Rules projects.
- [KG02c] introduces several variants of technical architectures of Business Rule systems, summarizes a criteria catalog for evaluating Business Rules products and gives a quick overview of several commercial products.

Main target audience for this document are business people that are not satisfied with their current IT systems as well as the responsiveness of their IT development departments. However IT consultants and IT developers may also gain insight about the notion of heavily business oriented IT systems.

1.2 Disclaimer

This document reflects the personal view of the author on the subject and represents is by no means the opinion of any community or vendor mentioned in the paper.

1.3 Definition of Terms

The following terms are used throughout the following chapters and thus require a unique definition:

Action Rule: A ⇨*Business Rule*, that automatically causes an action triggered by a manipulation of a ⇨*Business Object* or the occurrence of a ⇨*Business Event*.

Business Activity: A task motivated by a business objective (⇨*Business Motivation*). May be performed by either a ⇨*Business Actor* or an ⇨*IT-System*.

Business Actor: A (human) clerk within an organizational unit or an enterprise that has specific tasks and responsibilities assigned.

Business Event: An event (either happening outside of the context currently under investigation or a time event) that causes the execution of a ⇨*Business Activity*.

Business Location: A geographical place in which ⇨*Business Activities* are carried out by ⇨*Business Actors*. Also includes virtual places such as the Internet.

Business Motivation: A business oriented objective that serves as a motivation for a ⇨*Business Activity*.

Business Object: A business relevant, physical or abstract concept that is needed for carrying out a ⇨*Business Activity* or is produced or manipulated by a ⇨*Business Activity*.

Business Rule: A concise statement of a business relevant fact. One typically distinguishes between ⇨*Action Rules*, ⇨*Constraints*, ⇨*Guidelines* and ⇨*Derivations*.

Business Rule Engine: A technical software component, that is responsible for the efficient execution and monitoring of ⇨*Business* Rules.

Constraint: A ⇨*Business Rule*, that states a business relevant fact that must be true under any circumstances (i.e. whatever is done).

Derivation: A ⇨*Business Rule*, that defines, how desired information can be derived from given information. If this information is a numerical value, a derivation degenerates to a simple computation.

Guideline: A ⇨*Business Rule* that gives a suggestion to a human ⇨*Business Actor*, but does not cause any action by itself.

IT-System: A software system, that supports ⇨*Business Actors* as far as possible in carrying out their ⇨*Business Activities* within the constraints given by a set of ⇨*Business Rules*.

Use Case: A package of services that an ⇨*IT-System* provides to a ⇨*Business Actor* to carry out a specific ⇨*Business Activity*.

2. FOUNDATION

2.1 Today's IT-Systems

Would the car manufacturing industry be on the same level as today's information technology (IT), most people would not be able to drive a car themselves. A whole set of staff would be necessary to conduct the car in the direction we desire. If we could not afford to employ our own staff and/or if the resulting reaction speed (or better slowness) would not be sufficient for our transportation requirements, we would be forced to rely on public transport.

With exactly this situation we are faced with today's IT systems: companies own huge crowds of IT specialists that try to follow a changing business environment by providing supporting IT systems with a lag time of months or even years. However, smaller companies are usually not able to hire their own IT specialists and thus rely on standardized "off the shelves" software products that conform more or less to their requirements.

The Business Rules Approach (BRA) tries to address exactly this discrepancy between faster changing business requirements and an IT support that is constantly behind the business' needs. The approach pursues the visionary goal that not IT specialists but the business people themselves should be able to lead their business into the required direction. In other words: they should get a tool that allows them to build and adapt the required IT support themselves and immediately.

2.2 The Business Rules Approach

The Business Rules Approach (BRA) is a comprehensive approach that includes business engineering, IT systems development, Enterprise Application Integration (EAI) as well as technologies like rules engines, Web technologies or Application Servers. The Business Rules Approach is based on the following assumptions:

1. The business, not technology must be the driving force for IT development.
2. Business Rules are extremely valuable assets of an enterprise.
3. It must be simple (i.e. without specific IT knowledge) to "teach the business to a computer".
4. Business activities should be automated as far as possible, i.e. they should be carried out by an IT system.

To fulfill these requirements, the Business Rules Approach provides the following solutions:

- **Declarative instead of procedural description of the business by means of Business Rules**
  We should be able to „teach" the computer what to do, not how to do it. It should be the computer's task to decide how to achieve the goal.
- **Business Rules must be located in one single central place**
  Instead of distributing business logic among various places in the application code (or even bury it there), business logic will be placed in form of Business Rules in one single place within an application or even within the whole enterprise.

- **Specification of business rules that is oriented on natural language**
  In order to allow non-IT specialists to formulate Business Rules, they may not be expressed in a language comparable to a conventional programming language but they must be expressed in an English-like syntax (or any other natural language). Furthermore, usually tailored environments to define and maintain Business Rules are provided to business people to suit their specific needs and skills.
- **Integration into an existing IT environment (and thus the possibility for stepwise migration)**
  Since "big bang" replacements of IT systems today are no more realistic, Business Rules systems usually provide a large set of solutions for integrating legacy databases as well as application components.

2.3 Declarative Specification

Today's software implementation technologies are typically imperative or procedural: a program is a step by step description that tells the computer what is to be done and under what circumstances. In contrast, a declarative approach just specifies what to be done, i.e. the goal to be achieved, but not how. It is then the computer's task to decide about the best way to solve the given problem. Chris Date expressed this difference by the beautiful title of his Business Rules book "What Not How" [DATE2000].

Simon Williams [WIL2000] illustrates the same difference by a very common example. To describe a meeting point could be done in the following way: "*Take the motorway A1 towards Zürich and leave it at exit Zürich Altstetten. Turn right five times until you are on the bridge over the motorway. Then turn right again on the second traffic light and then go straight for about 1.5km until you reach the terminal station of the Tram. Continue straight for another about 500m and we meet in the restaurant at the left side on the road*".

The declarative version of this description would be much shorter: "*We meet in the restaurant at the Badenerstrasse 821 in 8048 Zürich.*" Beside the much more concise representation the declarative specification has the following advantages:

- The declarative description is still valid, when there are some road works or detour on the way.
- The declarative description is still usable, when I arrive from a completely different direction.
- The declarative description is implementation-independent: it is still usable when I walk to the meeting, or when I travel by taxi, train or airplane.
- The declarative description is easier to remember than the large circumstantial procedural version.

Certainly, declarative specifications not only have advantages: it requires a much larger self-sufficiency or even intelligence. Applied to IT systems this means: it will be more challenging for our computers... (but simpler for us humans).

2.4 Rules and Inferencing

In general, a rule has the following basic form:

```
IF premise THEN conclusion.
```

Depending on the given situation, we can apply such a rule in the following two ways:

- Forward Inferencing: If we know that the premise is true, we may assume the conclusion.
- Backward Inferencing: If we need to know the conclusion, we need to find out the premise.

The following guideline helps to decide about the appropriate Inferencing strategy:

- If we know what happened but don't know what to do, use forward inferencing rules to determine the actions to be done.
- If we are interested in a specific conclusion (or decision), but we don't know what information is required for this conclusion, use backward inferencing rules to determine the required premises.

Usually action rules are forward inferencing rules and constraints as well as derivations are backward inferencing rules.

2.5 Business and IT-Systems

To fulfill its mission, any enterprise carries out a whole set of business activities. These business activities are all motivated to achieve a common enterprise vision, but they are also constrained by the enterprise's environment as well as its business policies.

By introducing IT systems in an enterprise, its business activities will heavily be affected by those IT systems. The following three main influences of IT systems on business activities can be distinguished:

- Business activities may be supported by an IT system, i.e. business activities are simplified by means of IT services and the people responsible for those business activities become more efficient and produce less errors.
- Business activities may be completely automated by an IT system, i.e. there is no need for manual intervention to achieve a certain goal. These business activities are then performed in an extremely efficient way and in a very consistent manner.
- Business activities are not affected by an IT system at all. These are typically business activities that require a extensive human touch or very complex business knowledge that cannot be easily implemented in an IT system.

In any case, when an IT system affects a business activity, some business knowledge needs to be embedded into that IT system. This is where business rules become important.

3. BUSINESS CONCEPTS

3.1 The Zachman Framework

In the late 80s, John A. Zachman developed at IBM a framework [ZACH87] and [ZACHweb] for structuring information related to business and IT systems (see Figure 3-1). Today this framework also serves as a holistic basis for the Business Rules Approach.

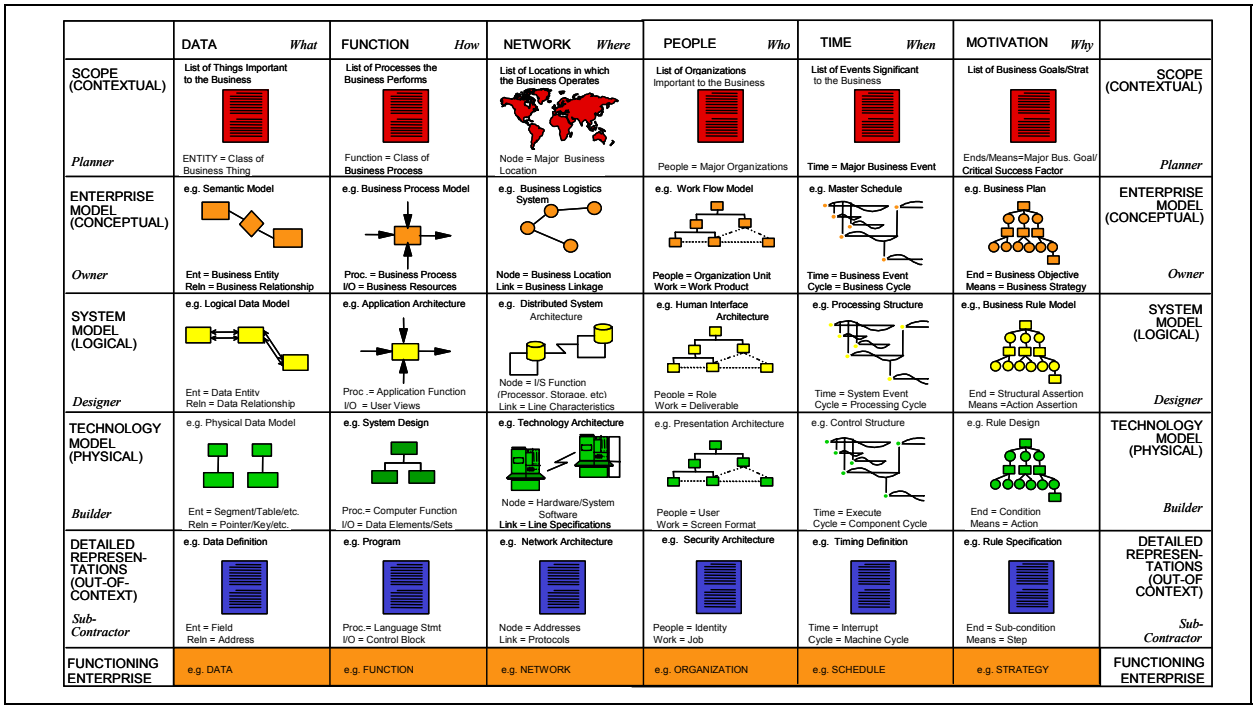

Figure 3-1: The Zachman Framework

The Zachman Framework organizes the description of business and IT systems into six orthogonal dimensions (represented as columns) relevant to six different stakeholders (represented as rows). The six dimensions (or columns) can be summarized as follows:

- **Column 1: What?**
  Describes the things that are important to the business and that might be represented in IT systems supporting the business. In other words: this column defines the *Business Objects*.
- **Column 2: How?**
  Describes the tasks to be performed in the business to achieve its objectives. These are also called the *Business Activities*.
- **Column 3: Where?**
  Describes the places where Business Actors execute Business Activities. These places are also called *Business Locations*.
- **Column 4: Who?**
  Defines the resources that execute Business Activities. These might be humans as well as IT systems but also more abstract resources such as organizational units. These resources are also called the *Business Actors*.

- **Column 5: When?**
  Defines the triggers for the Business Activities, i.e. the *Business Events* that cause Business Activities to start. This also includes time events.
- **Column 6: Why?**
  Describes he means and ends of the business or its reasons for its existence. These can be summarized as the *Business Motivations*.

Since the Business Rules Approach introduces the "WHAT not HOW" aspect of declarative specifications (see chapter 2.3), this causes a minor conflict with the headings of columns 1 and 2 of the Zachman Framework. For that reason we will introduce the following two slight modifications to the terminology of the original Zachman Framework:

- We will call column 2 (the functions) "WHAT" instead of "HOW".
- We will call column 1 (the data) "WITH" instead of "WHAT".

This minor modifications gives us six "W" columns: With, What, Who, Where, When and Why. The six stakeholders (or rows) of the Zachman Framework can be summarized as follows:

- **Row 1: Planner**
  Represents the perspective of a planner that defines the scope or context of a project within an enterprise.
- **Row 2: Owner**
  Represents the perspective of the owner of the business, i.e. makes the business explicit by describing it by an enterprise model on a conceptual level.
- **Row 3: Designer**
  Represents the perspective of the designer of an IT system, i.e. defines the boundary of such a system within the business as well as its specification in form of a logical system model.
- **Row 4: Builder**
  Represents the perspective of the builder of an IT system, i.e. defines its technology and architecture in form of a physical system model.
- **Row 5: Sub Contractor**
  Represents the perspective of the implementer of an IT system, i.e. describes detailed representations of its components needed and/or produced by individual sub contractors when they build these components. This information has typically a scope limited to a specific component and is thus usually out of context of the whole system.
- **Row 6: Executor**
  Represents the perspective of the functioning enterprise comprising of a set of established and running systems (mainly technical systems but also human systems) that actually perform or execute the business.

Note that there is a fundamental boundary between rows 2 and 3: rows 1 and 2 are purely related to the business, whereas rows 3 to 6 are related to IT systems that support the business.

3.2 Business Knowledge

All columns of row 2 in the Zachman Framework represent explicit information and knowledge about the business. But what is knowledge and what is the difference between information and knowledge? Figure 3-2 illustrates how we can progress from mere data to knowledge.

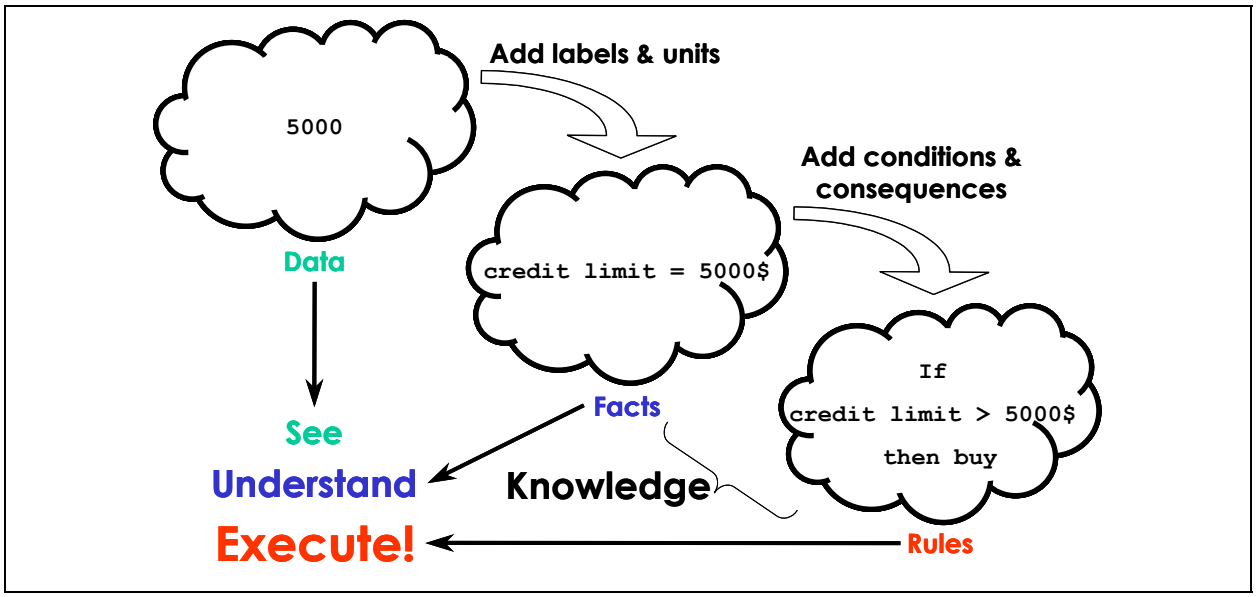

Figure 3-2: From Data to Knowledge

Figures without any given context are **Data**. Although we can see them, there is no way to derive any useful interpretation from simple data. By adding labels and units to figures, we can turn data into information or **Facts**. We now understand or interpret these figures and we may or may not react on such an interpretation depending on our intuition and/or experience. As soon as we add conditions and consequences to facts or derive facts from other facts, we know whether to react on a fact and how to react on it. We don't have to rely on our intuition; it is exactly defined what must be done in what situation: we gained insight defined by **Rules**. However, in order to understand we need to know the basic terms behind facts and rules. So, finally we can declare the following equation:

Business Knowledge = Business Terms + Business Facts + Business Rules

One of the main goal of the business Rules Approach is to make exactly this Knowledge explicit. This is what we need to know in order to run a business. The basic components of **Business Knowledge** are:

- **Business Terms**
  Terms that are important to the business such as customer or order. Business terms are usually represented as Business Objects.
- **Business Facts**
  Operational facts about the (current) business, also called business data. For example "Mr. Smith is currently owner of two open orders #371 and #438" is a business fact.
- **Business Rules**
  Generalized statements about business situations. For example "Any order that includes the product 'Guinness' is subject to a discount of 5%"

Changing and improving this knowledge over time is also called **learning**.

3.3 Business Motivation

The foundation to identify and manage any business should be a business plan. Beside other things, such a business plan should then be the source for identifying business rules. In other words: a business plan should make the motivation for business rules explicit. In 2000, the Business Rules Group [BRGweb] developed a document [BRG00b] that focuses on the systematic organization of such business plans. At the core of that document is a meta model that serves as a base for identifying and structuring key elements of any business plan. Figure 3-3 shows a simplified version of that meta model in UML notation.

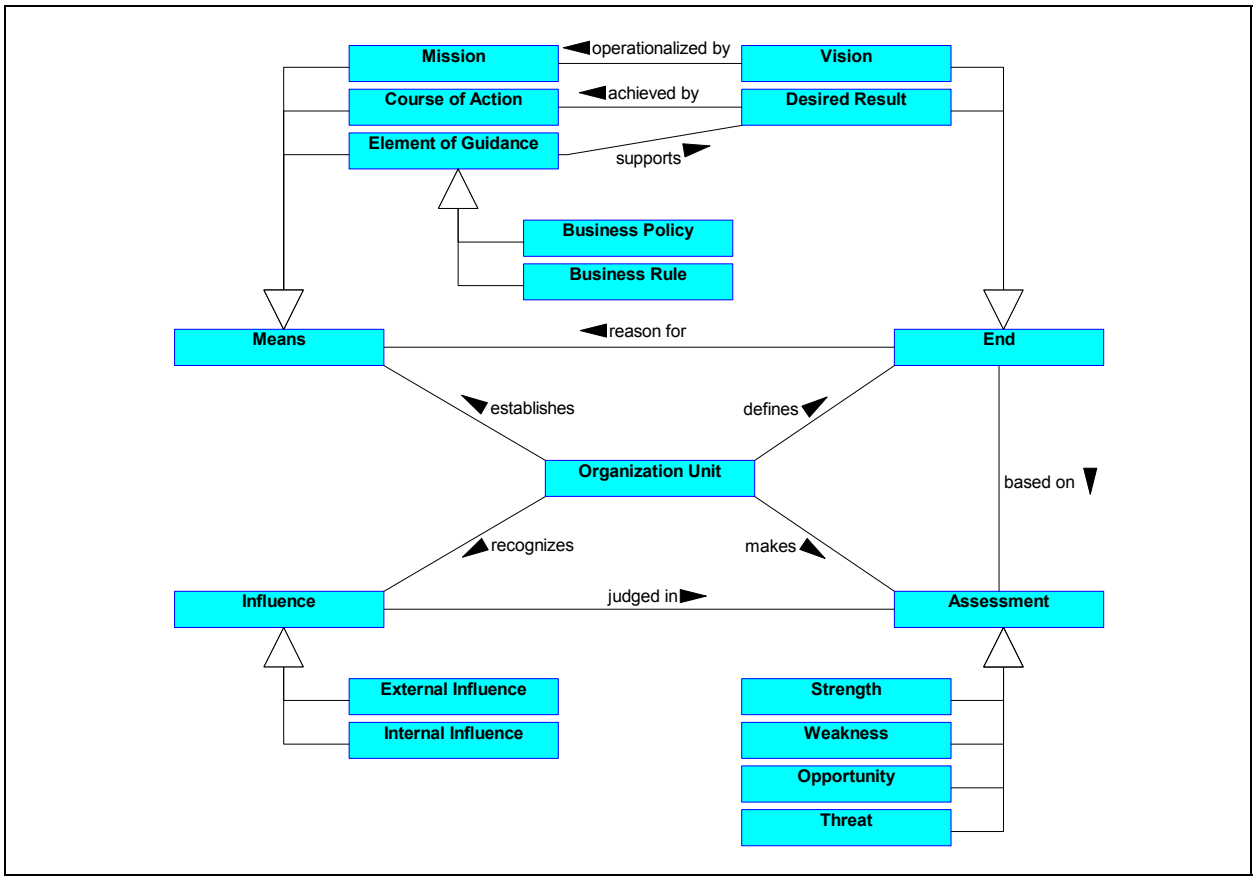

Figure 3-3: The Business Motivation Model (simplified)

The Business Motivation Model is based on the following four core concepts:

- **End**
  A statement of what is to be accomplished, such as a vision or the statement of a desired result.
- **Means**
  A device, capability, regime, technique, restriction, agency, instrument, or method that may be called upon, activated, or enforced to achieve Ends. In the context of this document specifically the Means "Business Policy" and "Business Rule" are of interest.
- **Influence**
  The act, process, or power of producing an effect without apparent exertion of tangible force or direct exercise of command, and often without deliberate effort or intent.
- **Assessment**
  A judgment that an Influence affects the employment of Means or the achievement of Ends.

3.4 Business Rules

As already mentioned in chapter 3.1, the Zachman Framework plays a central role in the Business Rules Approach. Figure 3-4 shows how each column of the Zachman Framework is related to the concept of a Business Rule:

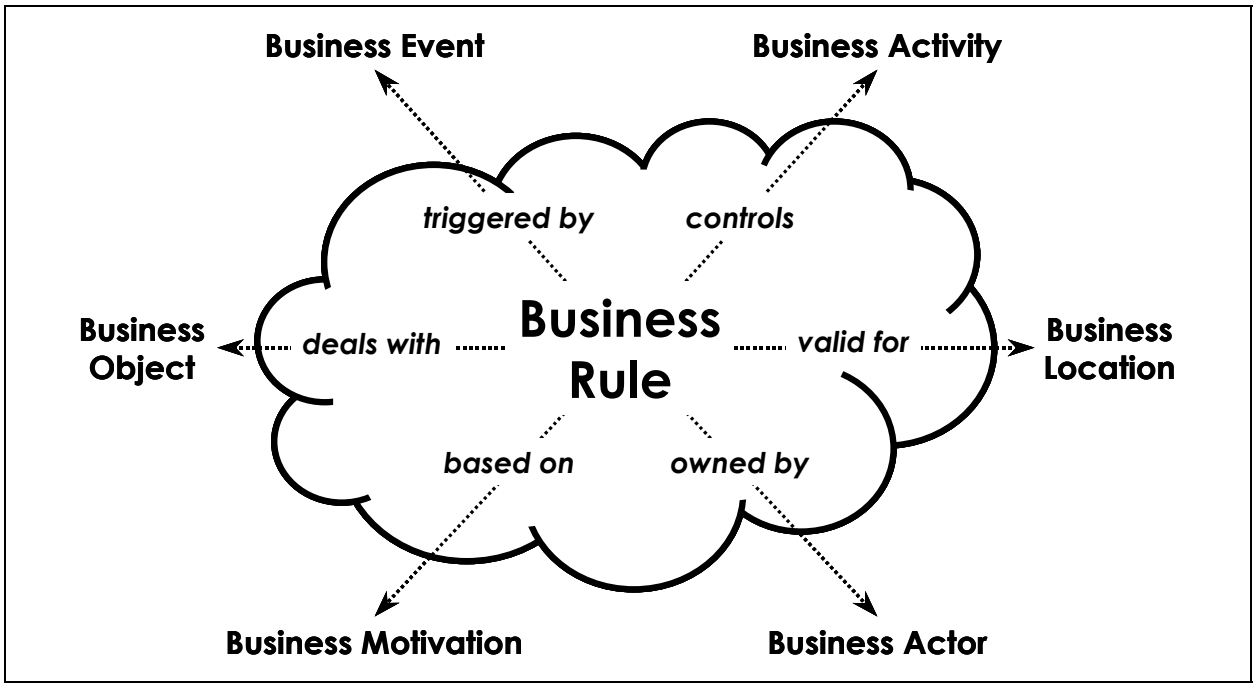

Figure 3-4: Business Rules & Zachman Framework

Behind every Business Rule should be a Business Motivation (column 6) that represents its reason for existence. For this reason it is highly recommended to clearly declare a Business Actor (column 4) to be the owner of any Business Rule. Business Rules deal with Business Objects (column 1) in two ways: they refer to Business Objects in order to determine the outcome of conditions and they often change Business Objects due to the rule's consequences. Any Business Rule (or set of Business Rules) is triggered by a Business Event (column 5) that occurred while a Business Activity (column 2) is performed or when such a Business Activity is triggered. Finally, since Business Rules often represent location-specific policies or even country-specific laws, some Business Rules may only be applicable or valid in certain Business Locations (column 3).

As we said earlier, Business Knowledge comprises of Business Terms, Business Facts and Business Rules. But what kinds of Business Rules can be distinguished? In literature there seems not yet a consensus about a classification scheme. For this reason, we use in this series of white papers the classification scheme shown in Figure 3-5 that represents a combination between [BRG00a], [DATE00a], [DATE00b], [HALLE02] and research at KnowGravity Inc.

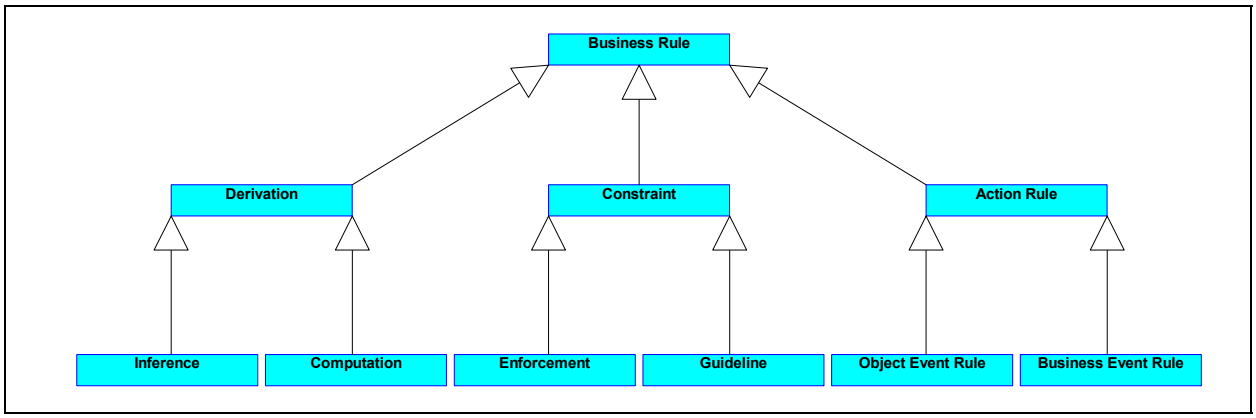

Figure 3-5: Types of Business Rules

Figure 3-5 above classifies Business Rules into the following three main types:

- **Derivations**
  Rules that derive "higher-level" information from other "lower-level" information. For example a customer's creditability is derived from it payment history, its current turnover, its age and possibly many other lower-level information. Derivations can further be specialized into Computations and Inferences: **Computations** derive numerical information (e.g. a credit limit) whereas **Inferences** derive non-numerical information (e.g. a customer's status). However, since both of these subtypes may be based on numerical and non-numerical information to evaluate their condition, this distinction may become blurred.

- **Constraints**
  Rules that limit possible actions within well defined boundaries. Any crossing of such a boundary will violate a constraint and cause some corrective actions. What such a corrective action causes depends on the type of the constraint: if it is an **Enforcement**, the action is completely rejected, if it is a **Guideline** a warning, possible with a required confirmation might be the more appropriate reaction. Constraints may also be use as validation rules to ensure for example that a form (paper-based or an application dialog) is filled out correctly.

- **Action Rules**
  Rule that cause constructive actions when a certain event occurs and/or when a certain condition is met. Such an action could be the manipulation of one or more Business Objects, the triggering of another Business Activity, the creation of an entry in the "to do" list of a Business Actor (Workflow rule) or any combination of such actions. Depending on the event that triggers an Action Rule, the following two distinctions can be made: **Object Event Rules** are triggered when a Business Object manipulation occurs (e.g. a new account is created) whereas **Business Event Rules** are triggered when a Business Event occurs (e.g. a money transfer is requested).

However, there are other classification criteria that produce orthogonal Business Rule taxonomies: for example the distinction between hard and soft rules, between explicit or implicit rules or between human guidelines or rules executed by a technical system. But for the moment we stay with the functional classification as shown in the diagram above.

3.5 Shared Enterprise Assets

Most people would agree that customers, employees, and money belong to the most valuable assets of an enterprise. Enterprises create dedicated and centralized functions such as "sales", "human resources", or "accounting" to manage those assets. These centralized functions serve the following purpose:

- They provide asset-related services with a single and consistent interface to the remaining enterprise as well as its environment.
- They reduce process redundancy in managing the asset.
- They minimize unhealthy internal competition in decentralized management.

In the 80's, data (or better: information) has also been identified as a highly valuable and shareable enterprise asset. This caused the creation of data management groups in many larger enterprises that where responsible for managing information as an enterprise-wide resource (see also chapter 4.2). The Business Rules Approach proposes to consider Business Knowledge, i.e. Business Information and Business Rules as a valuable asset of an enterprise. Table 3-1 shows the analogy between all these enterprise assets.

	Asset	**Goal**	**Department**	**IT-Support**
Environment	Customers & Relationships	Share customer associations among all departments of the enterprise	Sales	Customer Relation Management (CRM) Tools
	Suppliers & Partners	Provide reliable purchasing channels with good conditions to all departments of the enterprise	Purchasing	Purchasing and Partner Management Systems
Resources	Money & Inventory	Provide a well balanced financial background to the whole enterprise	Accounting	General Ledger Systems
	Employees & Skills	Maintain a loyal and well skilled base of employees of the enterprise	Human Resources	Skill Management Systems
Knowledge	Information	Share up-to-date business information (or business state) among all departments of the enterprise	Data Management	Information Repositories
	Business Rules	Provide a consistent business policy among all departments of the enterprise	Knowledge Management	Rule Repositories

Table 3-1: Shared Enterprise Assets

As introduced in chapter 3.2, Business Knowledge consists of the components Business Terms, Business Facts and Business Rules. As soon as an enterprise starts to make these items explicit (instead of having them implicitly as skills of its employees) and to share them among all its business, Business Knowledge becomes **Shared Business Knowledge**. And as soon as an enterprise starts to constantly update and improve this knowledge it becomes a **Learning Enterprise**.

Today, Business Terms are usually implicit in the heads of the enterprise's employees. The Business Facts are usually stored explicitly in large, enterprise-wide databases. However, Business Rules are today either implicit in the heads of employees or made explicit in some guidelines. In future, and that's the mission of the Business Rules Approach, they will become explicit (and automated) in Business Rules Systems.

4. IT-SYSTEM CONCEPTS

4.1 Anatomy of an IT-System

Today's IT-systems are usually built based on a layered architecture. Such a layered architecture increases the system's modularity as well as they encourage a concept called "separation of concerns": within a limited context, only a limited number of issues are relevant and must be considered. Figure 4-1 shows the typical layers of today's IT-System:

- **User Interface Layer**
  An IT-system provides its services to various different users (i.e. Business Actors) in form of **Use Cases**, typically started when a Business Event occurs. A use case is a package of fundamental IT-System services bundled to help users to carry out a Business Activity.
- **Workflow Layer**
  Use cases are typically started explicitly by a Business Actor, but they may also be activated from within the workflow layer. In this second case, the workflow layer decides which **Business Activity** must be carried out by which Business Actor and creates a corresponding entry in the "To Do"-list of that Business Actor. By opening such an entry, the corresponding use case is automatically started.
- **Business Object Layer**
  The fundamental services that are offered by means of use cases are provided by the **Business Objects** implemented in an IT-system. Business Objects incorporate the application or business logic and may interact with each other to implement business-conform functionality.
- **Data Storage Layer**
  Finally, the information represented by the Business Objects is persistently stored in **Data Objects** that are implemented in databases that are typically shared and accessible enterprise-wide.

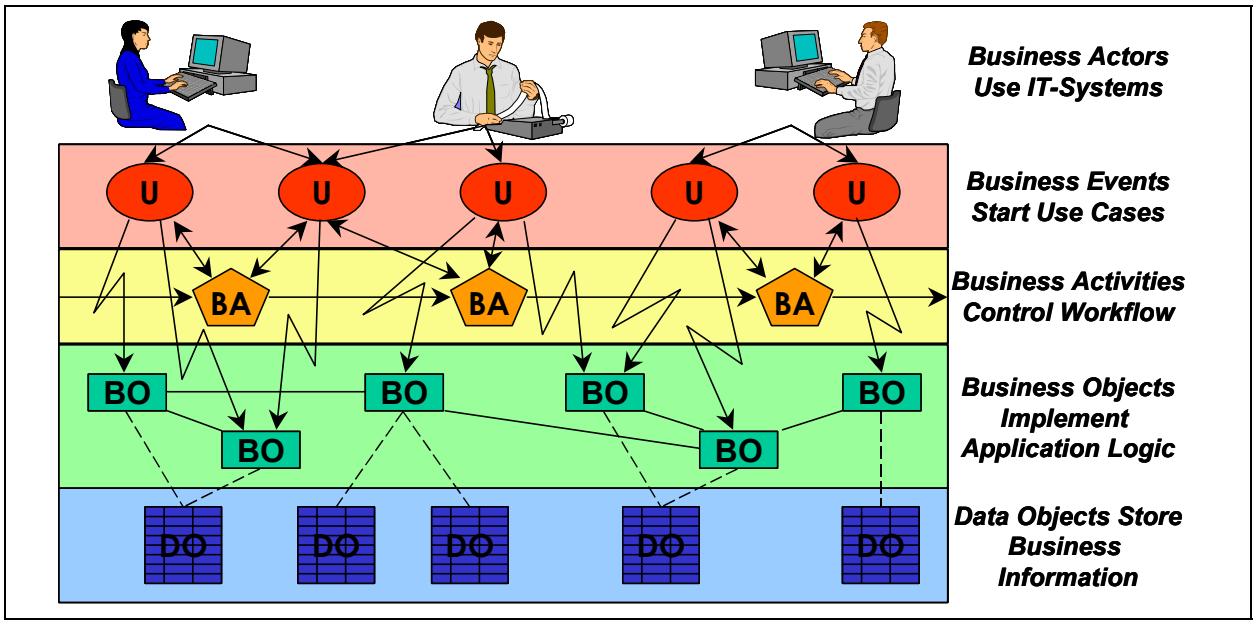

Figure 4-1: Anatomy of an IT-System

Today, these four layers of an IT-system are supported by various and different technologies as shown in Figure 4-2.

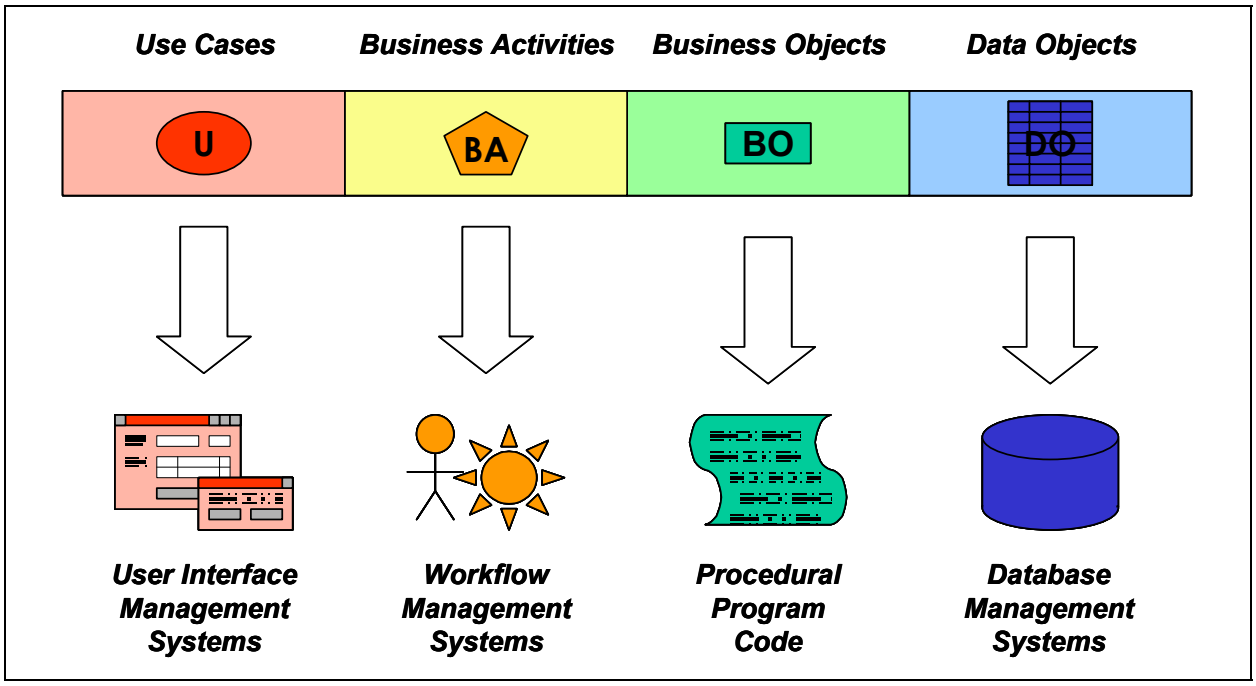

Figure 4-2: Supporting Technologies

The **User Interface Layer** of an IT-system is supported by windowing infrastructure such as Microsoft Windows or X-Windows derivates, but also by today's Web and Java technologies. Based on these technologies use cases with sophisticated (and hopefully convenient) user interfaces may be implemented to provide the IT-system's services to the user.

On the other hand, the **Data Storage Layer** is supported by many commercial database management systems (DBMS) that not only provide standardized access interfaces such as SQL or ODBC, but also capabilities to distribute data over multiple servers and locations. Furthermore, most of today's DBMS provide features that might be considered as initial functionality to support Business Rules: declarative constraints on columns and tables as well as stored procedures and triggers.

The **Workflow Layer** is supported by dedicated workflow management systems (WFMS). They too, usually have some limited rule capabilities built-in that allow the flexible and declarative definition of workflow based on simple decisions to forward a task to one of several possible Business Activities and/or Business Actors. These systems usually also provide means to interface with existing applications to activate the right use cases provided by these applications for the right Business Actors at the right time.

Finally, the **Business Object Layer** is today most often implemented in a procedural language such C, C++, Visual Basic or Java. In simpler systems the Business Object Layer is even merged with the User Interface Layer resulting in a mixture of code for handling the user interface with code that implements the application logic. However, if Java is the chosen implementation language, so called Application Servers are available that provide an Enterprise Java Beans environment that substantially simplifies common programming tasks such as DBMS access and transactions, security or load balancing.

4.2 Business Rules and IT Systems

The layered architecture introduced in chapter 4.1 provides several places for implementing Business Rules (starting from the bottom layer):

- In the **Data Storage Layer**, integrity constraints such as columns types and restrictions or referential integrity can be placed.
- In the **Business Object Layer**, computations and domain-level decisions in the interactions between Business Objects may be placed.
- The **Workflow Layer** typically contains rules about forwarding tasks to the appropriate Business Actors and trigger rules that activate other Business Activities.
- Rules for validating data entries are best placed in the **User Interface Layer**.
- And finally, rules that are not implementable or not yet implemented in an IT-system should be turned into guidelines for Business Actors.

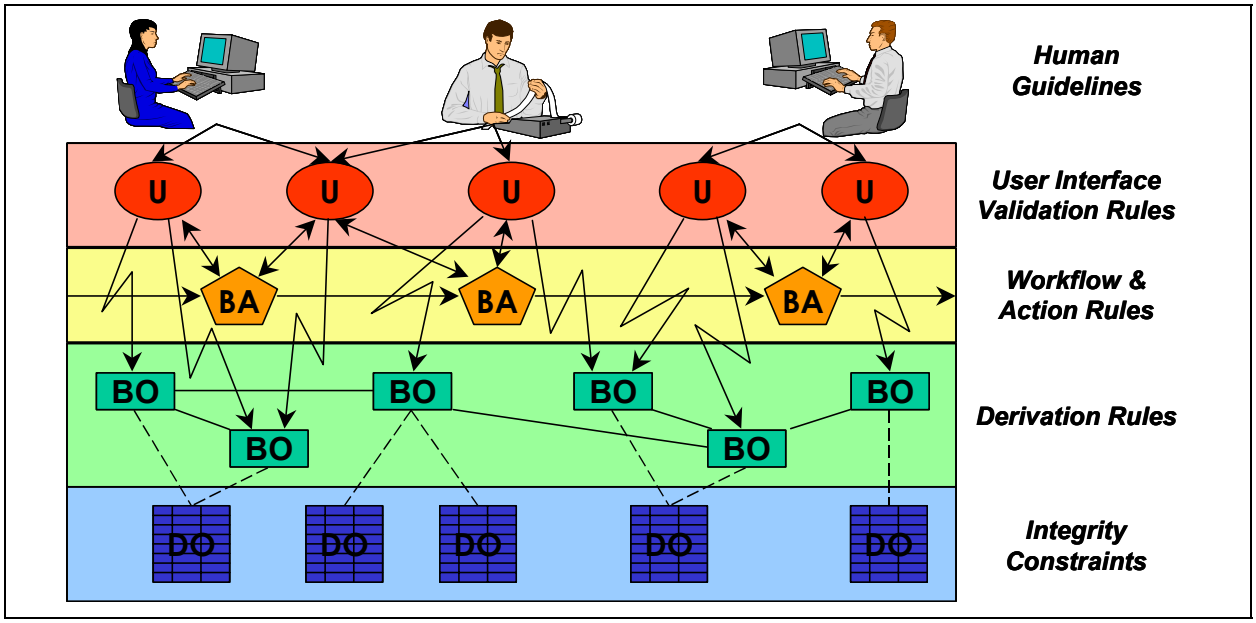

Figure 4-3: Places for Business Rules

However, when placing various types of rules into the four layers, a considerable overlap may occur:

- **Rules in the User Interface Layer versus Rules in the Workflow Layer**
  Decisions about the next Business Activities to be carried out can be taken on both of these layers. Typically, if they are implemented in the User Interface Layer, they are less flexible since they are hardcoded and deal with Business Activities relevant to the same (i.e. current) Business Actor. If these decisions are implemented in the Workflow Layer they are typically easier to change and deal with Business Activities across multiple Business Actors.
- **Rules in the User Interface Layer versus Rules in the Business Object Layer**
  Data validations may be carried-out in the User Interface Layer but also in the Business Object Layer. If they are implemented in the User Interface Layer, it is not possible for the user to enter incorrect data, since such errors are detected very early. If such validation rules are implemented in the Business Object Layer (only), incorrect data may be rejected after entry, which is less convenient for the user but results in more defensive Business Objects.

- **Rules in the Business Object Layer versus Rules in the Data Storage Layer**
  Constraints among Business Object can be implemented in the Business Object Layer but also in the Data Storage Layer. If they are implemented in the Data Storage Layer, they are as close to the actual data as possible and protect them as much as possible. However, today's DBMS are often limited in the support of general constraints and thus it is often easier to implement constraints in the procedural code of the Business Objects. The downside of this approach is that data is not protected if there is a possibility to access the data directly, i.e. without going through the Business Object Layer.

Deciding which types of rules should be implemented in which layer of the architecture is the main task of the Design phase in the Business Rules Approach. The lower rules are implemented, usually the more protecting and shareable (possibly even across multiple applications) they are. If rules are implemented in higher layers they are typically more powerful and may result in more convenient (and faster) user interfaces.

So one could devise an architecture that may implement some of the very same rules on more than one layer: on a higher layer for convenience and power and on a lower layer for protection and reuse. However, redundant rule implementations may lead to inconsistencies. As a consequence, Rules should be defined in one place but executed in as many places as necessary, which is the core proposition of the Business Rules Approach.

4.3 Evolution of IT-system Architectures

In chapter 3.5 we talked about information and rules as valuable assets of an enterprise. Based on the layered architecture introduced in chapter 4.1, it is interesting to see, how IT-systems evolved to reflect the recognition and support of those enterprise assets. Figure 4-4 illustrates these important steps in the evolution of IT-systems.

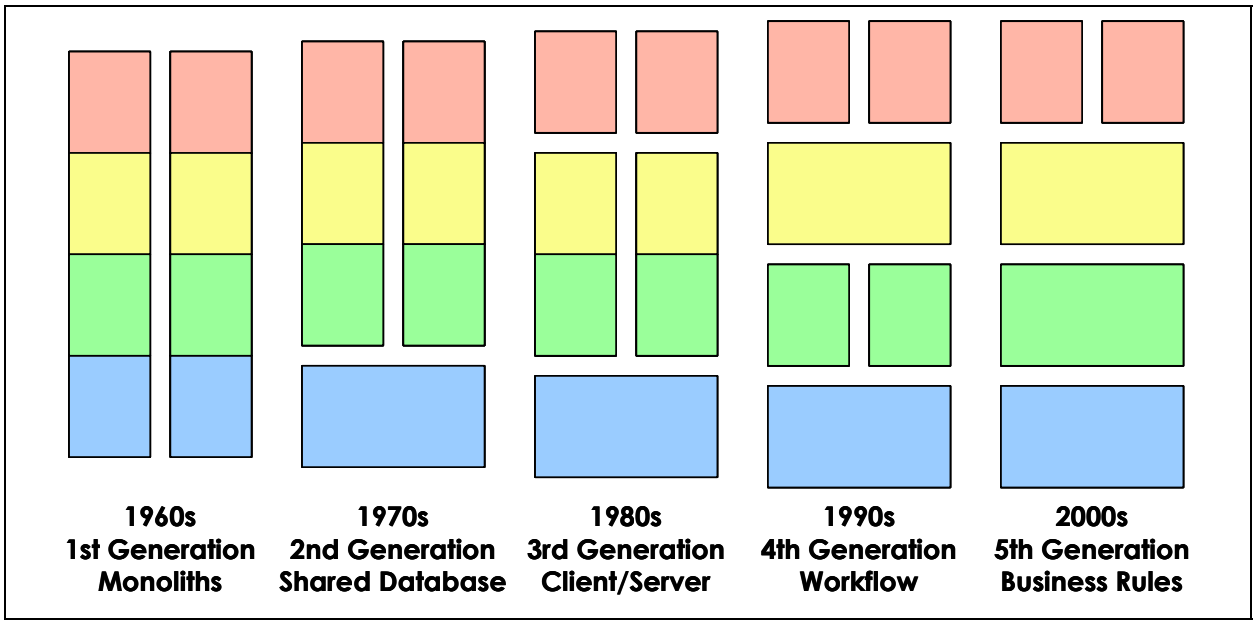

Figure 4-4: History of IT-Systems

In the **1960s**, IT systems where built as monolithical blocks that included everything from data storage to user interface and each application maintained its own local data. With the availability of the first database management systems in the **1970s**, it became possible to separate business data from the rest of an application and share it as a valuable resource across multiple applications.

With the introduction of the Client/Server architecture and the related technologies in the **1980's** IT systems moved from classical "dumb-terminal" applications to applications with sophisticated (graphical) user interfaces. These client systems provided their own intelligence and where based on dedicated and reusable user interface technologies such as windowing systems. This was also the advent of the today well-known "3-Tier Architecture".

When in the **1990s** business processes became the center of attention for IT systems development, dedicated Workflow Managements systems occurred on the market that allowed the integration of multiple and possible heterogeneous applications along a business process. However, the application logic remained still inside each application and there was no simple way to reuse this business-relevant (!) functionality across multiple applications.

With the start of the **new millennium**, the Business Rules Approach emerged: it is finally a way to share a common application logic across multiple applications and thus share the business knowledge as an enterprise asset in form of business rules.

4.4 The Business Rules Approach

The Business Rules Approach is a systems development methodology that has the Business Knowledge at the center of its focus. By "systems" we mean not only technical systems such as IT systems, but equally human systems such as enterprises. According to [HALE02], the following four "STEP" principles form the core of the Business Rules Approach:

STEP Principle	Motivation
Separate Rules	• To reuse rules • To apply special techniques to improve rule quality • To change rules independently of other system aspects
Trace Rules	• To determine, over time, if the rules remain correct for guiding the business • To assess the impact of rule changes
Externalize rules	• To allow everyone to know where rules can be known • To allow everyone to know what the rules are • To allow everyone to challenge the rules
Position rules for change	• To enable easy rule changes • To enable quick rule changes

Table 4-1: The STEP Principles

These four principles should always be kept in mind when a business or IT project is started. They ensure that the Business Knowledge is made explicit as well as maintainable.

4.5 Executable Rules and Rule Sets

On a generic level, any Business Rule (whether carried out by a Business Rules Engine or by a human actor) shares some properties illustrated by Figure 4-5. Each Business Rule is triggered (or activated) by an Event, which causes that the rule tries to gather information in order to make a decision. This may happen by inquiring Business Objects and/or by gathering additional information interactively from Business Actors. Depending on the outcome, an action is caused by the rule. Such an action may either be an action that affects some (other) Business Objects, or an action that produces a message to be sent to another Business Actor, and/or an action that triggers another Business Activity.

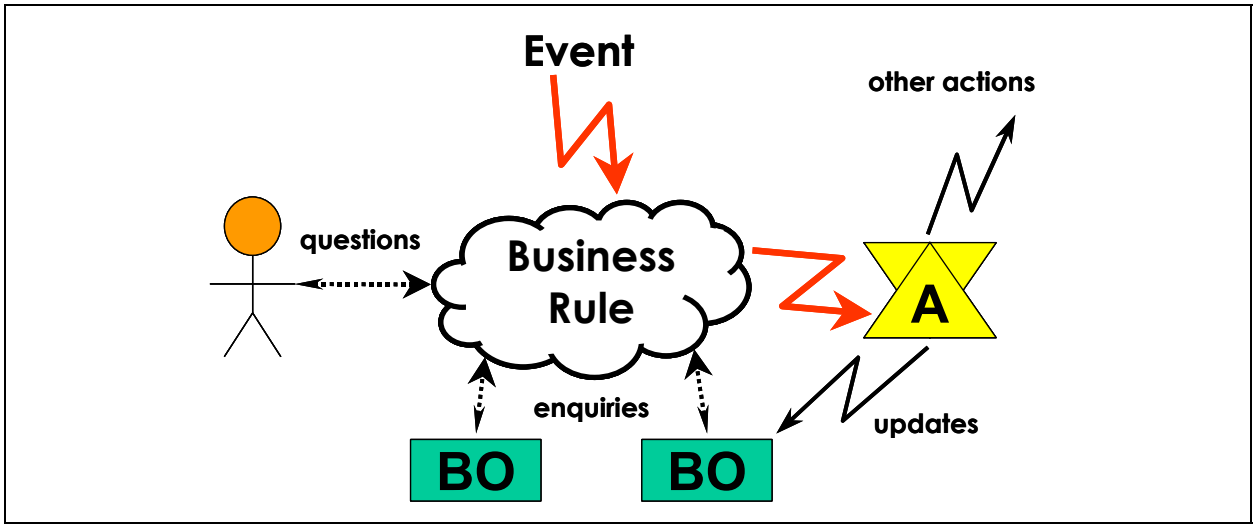

Figure 4-5: A Business Rule

When we compare this generic behavior of a Business Rule with the classification scheme introduced in chapter 3.4, the following table results:

Rule Type	Triggering Event	Performed Action
Derivation	Information request	Derive and deliver information
Constraint	Commit	Commit or rollback (or warning)
Action Rule	Business object change or Business event	Business object change and/or external action

Table 4-2: Business Rules Types

In contrast to the simple illustration above, Business Rules are usually grouped to whole sets of Business Rules contributing to a single decision or consequence. However, the basic mechanisms remain the same.

5. SUMMARY

The Business Rules Approach is a discipline for engineering business and IT systems with heavily relies on the notion of making Business Knowledge explicit. Conceptually it is based on the following four pillars:

- **Holistic approach that merges business and IT**
  IT should be considered as an integral part of the business and both should be regarded from different knowledge perspectives expressed by the six fundamental dimensions of the Zachman Framework: With, What, Who, Where, When, and Why.
- **Make IT adaptable to a changing business**
  Since the development of IT systems should be heavily based on the requirements of the business and the business constantly changing, technology should be provided that allows an easy and fast adaptation to new business requirements and policies.
- **Give power to the users**
  Business people should be able to conduct and control their business without the need to rely to much on IT people.
- **Declarative description**
  The notion of declarative descriptions of Business Knowledge should be in favor over procedural description since they are much more concise and leave more freedom in the way how a certain business goal is to be achieved.

In other words: the Business Rules Approach could be summarized as "**Business first!**". The business should be the driving force for developing any supporting technology and not the technology should force the business to adapt to the current capabilities of the technology. Business Knowledge must be incorporated into IT systems that will support the business, whereas Business Knowledge could be summarized by the following equation:

Business Knowledge = Business Terms + Business Facts + Business Rules

By making this knowledge explicit, shareable across an enterprise and easy updateable, we will finally achieve the vision of a **learning enterprise**.

APPENDIX A REFERENCES

[BRG00a] The Business Rules Group: *Defining Business Rules: What Are They Really? - Final Report, Revision 1.3*, July 2000, available at [BRGweb]

[BRG00b] The Business Rules Group: *Organizing Business Plans: The Standard Model for Business Rule Motivation, Revision 1.0*, November 2000, available at [BRGweb]

[BRGweb] www.BusinessRulesGroup.org: *The Business Rules Group*

[DATE00a] C. J. Date: *Twelve Rules for Business Rules*, 2000, available at www.aw.com/cseng/titles/0-201-70850-7

[DATE00b] C. J. Date: *WHAT Not HOW*, Addison-Wesley, 2000

[HALLE02] B. von Halle: *Business Rules Applied*, Wiley, 2002

[KG02a] KnowGravity Inc.: *Rules that Rule the Business - The Business Rules Approach: Applications*, 2002, available at [KGweb]

[KG02b] KnowGravity Inc.: *Rules that Rule the Business - The Business Rules Approach: Methodology*, 2002, available at [KGweb]

[KG02c] KnowGravity Inc.: *Rules that Rule the Business - The Business Rules Approach: Technology*, 2002, available at [KGweb]

[KGweb] www.knowgravity.com: *KnowGravity Inc.*

[WIL00] S. Williams: *The Associative Model of Data*, Lazy Software Ltd. (www.lazysoft.com), 2000

[ZACH87] J. A. Zachman: *A Framework for Information Systems Architecture*, IBM Systems Journal, vol. 26, no. 3, 1987

[ZACHweb] www.zifa.com: *The Zachman Framework for Enterprise Architecture*

