The Business Rule Approach

Eduard Bauer
University of Paderborn

Abstract. Today, a business faces fast changing environments which require frequent adaptations of parts of the business. These adaptations of the business are usually delayed to a great extent by the entailed modification of the business’s applications. Consequently, the business looses much time and money with adapting the applications’ behaviour and therefore cannot keep pace with changing environments. The modification deficiencies commonly result from the distribution of the business logic across the applications. The business rule approach tries to solve these deficiencies. It centralises the business logic and offers the business workers the possibility to modify the business logic on their own. Besides the analysis of the characteristics of the business rule approach, this paper focuses on the consequences of the business rule approach for the development process of applications, the implications on applications, and the advantages and disadvantages for the business.

1 Introduction

A business today faces several challenges: It has to cope with high competition pressure, upcoming new products and customer, changing customer preferences, dynamic environments, and so on. All these challenges require fast and rigorous changes of the business’s operations and decisions. In turn, these operations and decisions base on several constraints, rules, principles, and the business knowledge, which accordingly have to change as well.

One business uses several applications in order to support its operations and decisions. The parts of these applications that implement the constraints, rules, and principles are comprised under the term business logic. Therefore, the business logic is a foundation for the operations and decisions of a business [1].

Today, typically, the business logic is scattered over different applications [2], leading to redundant and inconsistent implementations of the same constraints and rules. Sometimes, business logic contains constraints and rules not explicitly known within the whole business, or the business assumes constraints and rules, not implemented in the business logic. Additionally, it is completely inaccessible by business workers and can only be modified by application developers.

This leads to several disadvantages in case of changing operations and decisions. Changing operations and decisions lead to changing rules, constraints, and principles and accordingly to changes of the business logic. Due to the distribution of the business logic across the applications, these changes require a lot of time and effort. Time and effort that has to be spent by special application developers, since the business logic is inaccessible to business workers.
These problems of distributed business logic that is not accessible by business workers are addressed by the business rule approach, which is introduced in the following section.

1.1 The Business Rule Approach

The *business rule approach* introduces *business rules* in order to solve the deficiency described above. A business rule is an atomic statement expressed by business workers about the rules, constraints, and principles of the business. More formally, the *Business Rule Group (BRG)* [3] specifies a business rule as follows\(^1\): ”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 behaviour of the business” [3]. In this case, the behaviour of a business can be seen as an aggregation of the business’s operations and decisions.

The business rule approach claims that all these business rules are collected and explicitly represented in a centralised application for the management of business rules. Such an application is called *business rule system* [4] [5] or *business rule management system (BRMS)*. Besides storing, it is in charge of managing and processing all the business rules.

*Implications of the business rule approach.* The introduction of business rules and the BRMS has several implications on the development process of applications, on the applications themselves, and finally on the business, where these applications are used.

The development process of applications used within the business changes in several dimensions. Without business rules, the different rules, constraints, and principles of a business are analysed and specified in the requirements specification. But, if the business follows the business rule approach, these business rules are already explicitly defined and stored within the BRMS. Accordingly, the business rules do not have to be recaptured in the requirements specification. Additionally, since rules, constraints, and principles are processed by the BRMS, they do not have to be implicitly implemented afresh, but can be used by the business’s applications.

This results in a central implication on the applications’ architecture. Along with the business rule approach, the distributed business logic is replaced by invocations of the BRMS or program code that is partly derived from the explicitly defined business rules. This leads to a separation of the business logic from the other parts of the business’s applications [4].

The main implication of the business rule approach on the business is that business workers are put in charge of creating and maintaining business rules. Accordingly, business rules are changed by the business’s workers without intervention of any application developer. This implies that the applications’ behaviour is adapted by business workers instead of application developers.

In the following, this paper depicts business rules in more detail in Section 2. Section 3 describes how business rules can be represented. The implement-
tation of business rules and BRMSs is depicted in Section 4. Finally, Section 5 presents the consequences of the business rule approach for the development process of applications, the implications on the business’s applications and their architecture, and the advantages and disadvantages for the business.

2 Explaining Business Rules

Since the most basic part of the business rule approach are business rules, this section describes, what business rules are, how they relate to other concepts of the business, the requirements for good business rules, and how such good business rules can be found.

In order to animate the abstract definition of the BRG and in order to describe the above parts vividly, two exemplary business rules are used. These are: ”An order is placed by at most one customer” (E1) and ”A customer that placed at least five orders is a golden customer” (E2). Both are expressed in natural English. But, while the first seems to express something like a constraint for orders, the second derives a status of a customer after a particular action of that customer.

2.1 Perspective

Business rules strongly relate to several concepts of the business. These relations can be seen in the layered structure shown in Figure 1. The following paragraphs explain the presented concepts and their relations consecutively, starting at the bottom. The colours used within Figure 1 differentiate between concepts and software.

Terms. The above exemplary business rules E1 and E2 make use of the word customer. Each business rule associates a particular standardised meaning with this word. Consequently, in order to avoid ambiguities and provide consistency for the whole business, these special words, called terms [6], have to be defined properly. All the terms and their definitions are comprised in a glossary, commonly named concept catalog [6]. These terms build a foundation for the other concepts, like shown in Figure 1.
Facts. In both exemplary business rules, the term customer relates to another term order. These relations between terms are called facts [6] or fact types [7]. Facts can set arbitrary many terms into relation. Facts do not define any constraints for these relations, but rather a general connection between terms. In this example a fact is for example order is placed by customer or a customer has a status.

All the terms and facts are comprised in the fact model [6]. The fact model generally describes all the business knowledge and consequently provides a common language that can be used by all business workers. It can be represented either textually or visually. In the latter case, terms are depicted as shapes, and facts as arcs between the shapes. One concrete state of the fact model is the instance model [6], which again might be presented textually or visually. The instance model can be seen as the situation currently faced by the business, with e.g. concrete customers, placing concrete orders².

Business Rules. At this point, business rules come into play. Business rules can now be seen as statements that constrain possible instance models, derive particular parts of the instance models according to a derivation instruction, or advice how the instance model should look like. Regarding the exemplary business rules, business rule E1 constrains the fact order is placed by customer, leading to the meaning that an order is placed by at most one customer. The second business rule E2 defines the modification of a particular part of the instance model, in case a customer performs a particular action.

Remaining Concepts. All these business rules are stored in the central BRMS. Building on business rules, business processes can be created [8]. The business processes can be seen as an additional technique to formulate the business logic found within applications. They can define the interconnection between several business rules from a higher point of view. Based on the BRMS and the business processes, the applications of the business are situated. Finally, the operations and decisions take all parts into account [1].

2.2 Categorisation

In order to handle the set of business rules within the business, business rules are usually classified according to several characteristics. Although a lot of different classifications according to completely different characteristics exist, I will give only one general distinction between three different categories of business rules. This distinction bases on the layered structure given afore in Figure 1 and the relations of business rules and instance models.

² This concept relates strongly to the technical concept of a domain model and instances of this domain model. In this case domain models might be represented by UML class diagrams and instances of this domain model by UML object diagrams. Nevertheless, there are several differences between fact models and domain models, respectively instance models and instances of domain models. The most important difference is the strong relationship of the fact model and its instance models. A more detailed analysis of these relationships can be found in [6].
Regarding the impacts of business rules on the instance model, three general categories of business rules can be derived. The first category constraints possible instance models. Thus, such business rules are checked, in case of a modification of the instance model. In case that some business rules are not fulfilled after the instance model has been modified, the according actions are rejected. Consequently, this category is called rejecter [6]. The business rule E1 belongs to this category. The second category derives or produces additional pieces of information from the instance model. Thus, this category is called producer [6]. An example business rule that belongs to this category is E2. The third category is much less restrictive than rejecter and producer. This category contains business rules that describe general constraints and derivations that do not always have to be fulfilled, but can rather be seen as advices/facultative business rules. Accordingly, this category is called advice [7].

2.3 Requirements

Nearly every statement concerning a business can be regarded as a business rule. One challenge, the business is facing along with the business rule approach, is to find good business rules. The BRG collected and summed up several requirements for business rules [9], which lead to good business rules. Supplemented by some additional requirements [1], the following list of requirements results:

1. **atomic**: can’t be broken down any further without losing information
2. **business related**: only use terms and facts of the fact model
3. **consistent**: a business rule does not contradict another one
4. **declarative**: no procedural description
5. **unambiguous**: have only one, obvious interpretation

The exemplary business rules E1 and E2 obviously fulfil all these requirements, if we assume that the underlying facts and terms are properly defined within the concept catalogue and the fact model.

2.4 Discovery

A complex part of the business rule approach is the discovery of business rules. The goal of the discovery process of business rules are a set of good business rules, stored in and managed by the BRMS. The discovery might on the one hand be done along with the development of an application [4] or on the other hand independent of any developed application. In the former case, the discovery becomes one step of the development process of an application and is therefore comparable to the requirements specification. However, in contrast to the requirements specification, the discovery of business rules concentrates completely on business rules and leads to a central BRMS filled with business rules. In the latter case, the business decides independently of any application in development to follow the business rule approach and to introduce a central BRMS.
**Origins of Business Rules.** In order to discover business rules we need to know, where business rules can originate. In general, they can originate either *internally*, this means within the business, or *externally*, this means independent/apart from the business [1]. Internal origins might for example be business principles, the business philosophy, or other enforcements present within the business. For example, the exemplary business rule E2 might originate in a business principle that favours loyal customers. In contrast, external origins are enforcements which usually result from laws, quality standards, or other external restrictions.

In order to disclose business rules, the three different sources *people, documents*, and *code* [4] are used. Several fine grained techniques, like *interviews, workshops, walkthroughs, and inspections* exist [1], which are not further investigated in this paper.

### 3 Representing Business Rules

After business rules have been discovered, they must be represented in a defined way. A way that is more defined than usual natural language, without ambiguities and obscurities. Accordingly, special languages, called *business rule languages*, have to be used. In general any defined language that can express constraints or rules is conceivable for business rules. But, resulting from the request of the business rule approach that business rules must be expressible by business workers, the main requirement towards these languages—the ease of use—must be fulfilled.

This section introduces two in many aspects differing languages for conceptually representing business rules and analyses characteristics of these languages. The two analysed languages are *RuleSpeak Business Rule Notation (RS)* [6] and *Object Role Modelling 2 (ORM)* [10], [11]. I have chosen these two languages due to their different characteristics and their fulfilment of the requirement—the ease of use.

#### 3.1 Representation

In general, two opposing representations of business rules are distinguished, *textual* and *visual* languages. While RS is a textual language, business rules composed in ORM are represented visually. Table 1 contains the two exemplary business rules E1 and E2 expressed in the two different exemplary business rule language.

**Textual Languages.** The first kind of languages, textual languages, usually strongly resemble natural languages, like for example natural English. Nevertheless, they constrain these natural languages to defined sentences, which make use of special words. This means they introduce predefined *patterns*, which use predefined *keywords*. While keywords are special words like IF, MUST, SHOULD\(^3\), and so on, patterns define the usual structure of the sentences. One such pattern is for

---

\(^3\) These words are usually printed in capital letters
Exemplary Business Rule E1
An order is placed by at most one customer (E1)

Exemplary Business Rule E2
A customer that placed at least five orders is a golden customer (E2)

<table>
<thead>
<tr>
<th>Natural Language</th>
<th>RuleSpeak</th>
<th>Object Role Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>An order is placed by at most one customer (E1)</td>
<td>An Order MUST NOT be placed by more than one customer.</td>
<td>![Object Role Modeling Diagram]</td>
</tr>
<tr>
<td>A customer that placed at least five orders is a golden customer (E2)</td>
<td>The customer status MUST BE SET TO golden, WHEN the customer has placed five orders.</td>
<td>* Customer has GoldenStatus iff each Order which is placed by Customer &gt;= 5</td>
</tr>
</tbody>
</table>

Table 1. Representation of the exemplary Business Rules

example the Event Condition Action pattern [12]. In general, this pattern has the following structure: ON event IF condition DO action. In this case, the event, the condition and the action are formulated in natural language. The common usage of natural language for parts of the patterns results is an inherent disadvantage of textual languages. Although they define patterns and keywords, they remain ambiguous and obscure. Such ambiguities are introduced by words like some, generally, and mainly, which might be used in parts of the patterns.

RS defines several different patterns in order to express business rules. This language assigns its patterns to categories, related to the three categories defined in Subsection 2.2. The pattern used for representing the exemplary business rule E1 makes use of the keyword MUST NOT and accordingly belongs to the rejecters.

Visual Languages. In contrast, visual languages represent business rules by different interconnected visual constructs. Such constructs might for example be facts and terms of the fact model. A variety of different visual constructs and notations are conceivable for business rules. Some of these notations were already compared by Herbst et al [2]. The paper at hand only gives a brief insight into one possible visual language for business rules—the ORM.

ORM, originating in the database domain, aims to facilitate modeling, transforming, and querying business information. For that the language provides constructs for modeling the fact model and extending this fact model by business rules. ORM represents terms by object types, symbolised by soft rectangles. Facts are represented by roles, visually presented by several connected hard rectangles. The number of the hard rectangles corresponds to the arity of the fact. Additional visual constructs, like the lines above the boxes, constrain the fact model, and can accordingly be used to represent business rules. The line above the fact is placed by expresses that each order is placed by at most one customer, resulting in exactly the meaning of the exemplary business rule E1.

3.2 Expressiveness/Completeness
Another requirement towards business rule languages is rich expressiveness. This means to which degree the business rules that are expressed in natural language
can be expressed in the according business rule language. Zur Muehlen et al. [13] analysed the expressiveness of several business rule languages in a formal way. They based their analysis on the Bunge-Wand-Weber ontology [14] and investigated whether the compared languages are capable of expressing constructs of the underlying ontology. Accordingly, they analysed the ontological completeness of these languages. In contrast, this paper analysis the expressiveness of the two exemplary business rule languages with regard to the three categories of business rules defined in Subsection 2.2.

Analysis of Expressiveness. Remember that RS assigns the different possible patterns to categories that resemble the defined categories in Subsection 2.2. Accordingly, all kinds of business rules can be composed in this language. Examples for business rules belonging to the first two categories, composed in RS, are shown in Table 1. An example pattern for the last category, the category advice, makes use of the keyword SHOULD, which indicates optionality.

Examples for business rules which belong to the first two categories expressed in ORM can be found in Table 1. In contrast to RS, ORM is not capable of expressing business rules of this last category. A reason for this can be seen in the origins of ORM—the database domain—where such facultative expressions are not usable.

3.3 Formality

Languages are based on different concepts that define how sentences or models are composed in that language. Formality addresses the degree to which these concepts are formalised. Formal language definitions wipe out ambiguities and therefore support the requirement unambiguous of business rules defined in Subsection 2.3. Nonetheless, formal languages tend to be complex and complicated.

Only parts of RS are formalised. All the keywords and pattern are defined partly formally. But, due to the usage possibility of natural language within the patterns, the language RS remains ambiguous.

ORM, in contrast, has a well-defined mapping to formal logic [15]. Accordingly, the different constructs of this language have a defined meaning, leading to unambiguous business rules.

3.4 Usability

Since one main requirement of the business rule approach towards business rule languages is the ease of use for business workers, the according business rule language must be simple, rapidly learnable, and easily usable. By that, the business rule language supports the business communication between the business workers and leads to comprehensible mutual understanding.

All the aspects of business rule languages described above, like representation, expressiveness, and formality affect the usability of the business rule language. Representation. Textual languages tend to be easier usable than visual languages. Nevertheless, the different pattern and keywords and their usage have
to be known in order to create correct textual business rules\(^4\). Visual languages can provide a fast overview over a business rule, the different visual constructs have to be learned and understood afore. Regarding ORM, an advantage of this business rule language is that it uses the same visual constructs for the definition of the fact model and the business rules. But, this can cause a mixture of the fact model and business rules.

**Expressiveness.** Expressive languages provide several akin constructs in order to express the same business rule in different ways. Accordingly, such languages increase the usability, since the one and only suitable construct does not have to be kept in mind, but other constructs can be used more intuitively. Less expressive languages, like ORM, which do not provide constructs in order to express a whole category of business rules, reduce the usability of the language to a great extent.

**Formality.** Finally, investigating the impact of formality on usability, formally defined business rule languages tend to decrease the usability and rapid learnability. Due to the formal definition, only few constructs are permitted, while prohibiting several arbitrary constructs. Regarding the two exemplary business rule languages, the possibility of RS to formulate parts of the patterns in natural language leads to an easier usage of that language, at the expense of a decrease formality.

## 4 Implementing Business Rules

After the discovery of business rules, they are specified by business workers with the help of textual or visual languages. Usually, these representations of business rules are not automatically processable by the business’s applications. Accordingly, the business rules have to be translated to another representation which is based on a formal representation, and is therefore automatically processable. These representations tend to be not legible and usable by business workers.

**Processable Representation of Business Rules.** Such a formal automatically processable representation of business rules is provided by the *Semantics of Business Vocabulary and Business Rules (SBVR)* [7] defined and standardised by the *Object Management Group (OMG)* [16]. SBVR comes along with a meta model that specifies formally, how terms, facts, and business rules are expressed. The meta model itself is aligned to the first order predicate logic.

Besides the definition of a meta model, SBVR describes how several business rule languages can be mapped to its meta model. Such a mapping is defined for both exemplary business rule languages, RS and ORM.

**The BRMS.** This translation from human readable business rules to processable business is one task of the BRMS. Besides this translation, the BRMS is capable of storing, managing, and processing all the business rules collected within a business. Accordingly, the BRMS is the linchpin of the business rule approach.

---

\(^4\) Although it might take some time to find the correct pattern in order to express a business rule previously formulated in natural language.
The general structure of a BRMS is shown in Figure 2. In this figure, the BRMS is shown on top of a typical three layer architecture of a business application. This application uses the BRMS to achieve its functionality.

![Figure 2. Business Rule Management System](image)

On the whole, the BRMS consists of three major parts [5]. The first part are several tools, used by business workers to define business rules. These business rules are stored within the second part, the business rule repository and used by the last major part, the business rule engine. The business rule engine is invoked by applications via defined interfaces and interprets the stored business rules. Besides the interpretation of business rules, the BRMS can provide a possibility to generate code which represents business rules [1]. This leads to the direct execution of business rules.

Such code or the functionality of the business rule engine can be seen as the centralised business logic invoked by the business’s applications, as shown in Figure 2.

5 Discussing Business Rules

Having defined the business rule approach, how business rules are used, and how they are represented let us focus in this section on the implications, the advantages, and disadvantages of the business rule approach. Recall that the main two implications of the business rule approach are the introduction of business rules, which are expressible by business workers, and the usage of a central BRMS to store, manage, and process these business rules.

In this section, the advantages and disadvantages\(^5\) are classified according to three different aspects, found within businesses, which are strongly affected by the business rule approach. These are firstly the development process of applications, including the introduction of a central BRMS, secondly the business’s applications, and thirdly the business’s workers, decisions and operations. Analogously, this section is divided into three sub-sections which separately deal

---

\(^5\) The advantages and disadvantages were comprised from the references [1] [4] [6].
with these aspects. Each sub-section closes with a table that summarises all the given advantages of disadvantages. Throughout the following sub-sections the implications on the scope-triangle [17], consisting of cost, time, and quality are analysed.

5.1 Development Process

If an application is intended to be developed according to the business rule approach, but the business does not have a centralised BRMS, filled with good business rules, the setup of the BRMS and the collecting of business rules can be done in the course of the development process of the application. Accordingly, let us first concentrate on the setup of a BRMS.

*Introduction of the BRMS.* The first step towards the business rule approach is the introduction of BRMS. This setup process is one of the greatest disadvantages and challenges for a business, regarding time, money, and quality. Within this process, a BRMS has to be set up, all business rules collected and discovered [6], and all existing applications adapted. This results in a significant investment, amount of time, and risk for the business [1]. On the contrary, once set up, such a BRMS leads to a continuous process in which additional business rules might be added easily and incrementally [4].

*Development process of new applications.* The presence of such a central BRMS, yields reduction of needed time and money for the development of new applications. Since all business rules are stated explicitly in the BRMS, they do not have to be collected and defined in the requirements specification. Additionally, the business logic of the application in development does not have to be created and tested afresh, but can be replaced by invocations of the BRMS. Accordingly, the development process is shortened and cheapened [4].

The advantage of generating code from the BRMS, like stated by Morgan et all. [1], applies only under exceptional circumstances. The generation of code for all business rules seems rather unrealistic. The reason for this is that this would require that business rules are not formulated abstractly, but very close to program code.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>− Continuous and incremental addition of</td>
<td>− Introduction of BRMS</td>
</tr>
<tr>
<td>business rules to BRMS possible</td>
<td>• Significant investment</td>
</tr>
<tr>
<td>− Less expensive and time consuming</td>
<td>• Large amount of time needed</td>
</tr>
<tr>
<td>development of new applications</td>
<td>• High risk for business</td>
</tr>
</tbody>
</table>

 Table 2. Advantages and Disadvantages regarding the Development Process

5.2 Business’s Applications

Following the business rule approach, the application landscape of a business comprises the central BRMS and all other applications that use the BRMS in order to achieve their functionality. Accordingly, this subsection first investigates the advantages and disadvantages of the central BRMS, and then focuses on the implications of the business rule approach on other applications.


Advantages and disadvantages of the BRMS. Due to the centralisation, a defined system access point within the business results, where business rules are homogeneously accessed and changed [4]. Each business rule is expressed only once and used by all applications found within the business. Consequently, changes to one business rule result in changes to the whole application landscape, yielding time efficient and cost effective change management.

But, on the contrary, the centralisation entails several disadvantages [4], especially due to the strong dependence of all the applications on the functionality of the BRMS. The single point point of access and management leads to a single point of failure. Breakdowns of the central BRMS result in breakdowns of the business’s application landscape. Accordingly, the whole business stops operating, causing large financial damages. Additionally, the central BRMS—interpreting the whole set of business rules—can become a bottleneck for all applications. Supplementary computers for the BRMS might be useless, since centralised applications usually tend to scale badly.

Along with the introduction of one BRMS, the business relies on one vendor of such BRMSs [1]. Accordingly, the business depends heavily on this vendor in case of failures and updates of the BRMS. If the vendor ceases to work on the BRMS, the business has either to stick with the old BRMS, or has to migrate to another BRMS. In the latter case, this might lead to a large financial burden for the business.

Besides, one functionality provided by BRMSs is crucial for the whole business rule approach. This is the translation of easy expressible business rules to logical, consistent, and unambiguous representations of these business rules [1]. In case of failures or misconceptions within these translations, the applications behave differently compared to their intended behaviour.

Furthermore, one additional functionality of the business rule management system must be the detection and disclosure of inconsistent and contradicting business rules within the business rule repository. Techniques, how this can be achieved, are rarely given along the business rule approach.

Implications on other applications. Regarding the implications on the applications that use the central BRMS, especially the software quality—in particular modifiability—of these applications is increased in several ways.

The split between the BRMS and the other applications, entails the clear separation of the business logic and other parts of applications like data and presentation. This increases the ease of modification. Especially, the exchangeability of whole applications, while preserving the same business logic becomes possible.

Besides, due to this split, the BRMS and the other applications can evolve independently, allowing the usage of arbitrary technologies for the different applications. This fact, in turn, increases the modifiability of the business’s application landscape.

The centralisation of the whole business logic, leads to the abolishment of the distributed business logic, scattered across several applications. This yields consistency among all applications used within a business. Additionally, since each
business rule is stored centrally and used by several applications, modifications of each business rule lead to fast adaptations of several applications.

Besides, the centralisation tends to decrease the size of applications. This yields less opportunities for errors and bugs, resulting in less error prone applications. Less error prone applications are usually easy to maintain, which leads to savings of cost and time. Besides, the time and cost for testing the application decreases along with the decrease in application’s size.

Nevertheless, opposing these advantages, the additional usage of the BRMS tends to decrease the performance of the other applications. Every invocation of the business rule engine results in an additional overhead in time and computation.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business rule management system</td>
<td>Business rule management system</td>
</tr>
<tr>
<td>Homogeneous access for business rules</td>
<td>Single point of failure</td>
</tr>
<tr>
<td>Reusing same business rules across application landscape</td>
<td>Bottleneck of whole application landscape</td>
</tr>
<tr>
<td>Time efficient, cost effective adaptation of the applications behaviour</td>
<td>Bad scalability</td>
</tr>
<tr>
<td>Other applications that use the BRMS</td>
<td>Dependence on specific vendor</td>
</tr>
<tr>
<td>Increased modifiability</td>
<td>Translation misconceptions possible</td>
</tr>
<tr>
<td>Use of arbitrary technology</td>
<td>Disclosure of inconsistencies among business rules necessary</td>
</tr>
<tr>
<td>Behaviour consistency among applications</td>
<td>Other applications that use the BRMS</td>
</tr>
<tr>
<td>Decreased size of applications</td>
<td>Heavy dependence on BRMS</td>
</tr>
<tr>
<td>Less opportunities for errors</td>
<td>Decreased performance</td>
</tr>
</tbody>
</table>

Table 3. Advantages and Disadvantages regarding the BRMS and Applications

5.3 Business

In this subsection the advantages and disadvantages of the business rule approach are investigated concerning the business. For that purpose, we analyse the implications on business workers and the potential adaptability of the business’s operations and decisions in the following paragraphs.

Implications on business worker. One of the main requirements of the business rule approach towards business rules is the possibility to express business rules by business workers. Accordingly, the main advantage for business workers is the possibility to express business rules on their own. Consequently, business worker have to be able to work with the BRMS and express their own business rules. Usually, this is not possible without any education and training of the business workers. The education of all business workers to use such systems requires a large amount of money and time [1].

Additionally, the mandatory usage of the BRMS results in an extra task for the business worker. This imposes more work and responsibility resting on the shoulders of the business workers.

Although one of the requirements of business rules is unambiguity, this is nearly not achievable with languages akin to natural languages (cp. Section 3). Consequently, some ambiguities might remain within the BRMS—especially when
used by several different business workers, with different knowledge and general understanding.

Another disadvantage for business workers is the explicit definition of the business rules within the BRMS. This implies that business workers do not have any implicit business knowledge that makes them important for the business. Because of this, businesses are not bound as tightly to their business workers as earlier, leading to unproblematic exchangeability of business workers.

**Adaptability of the business.** The business rule approach aims especially to improve the adaptability of the business’s operations and decisions. Recall that the main idea to improve this adaptability is to remove the implicit business logic caved within the distributed applications, and to add explicitly defined business rules to the BRMS. Accordingly, the business does no longer depend on external application developers to adapt application behaviour, but gets in charge of its own destiny. Accordingly, the business’s applications remain no barrier for the adaptation of the operations and decisions of the business [4].

If business rules are expressed in a standardised way—like for example with the help of SBVR (cp. Section 4), two advantages result. Firstly, new legislative mandates and laws might be directly expressed as a set of business rules. Assuming the same fact model, this enables the possibility to share the set of legislative business rules among several businesses, exploiting the easy adaptability of the businesses. Secondly, mergers and acquisitions become easier, if both businesses make use of the same representation of business rules and maintain similar fact models [4]. In this case, it is conceivable to integrate all the business rules of one business into the BRMS of the other business. Nevertheless, this can result in a great challenge, consuming large amount of time and money in order to remove inconsistencies and redundancies.

Additionally, the business can improve the individual customer support. This results from the easy adaptation of business rules by business worker which directly get in touch with the individual customers [6].

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>− Business worker</td>
<td>− Business worker</td>
</tr>
<tr>
<td>● Able to express business rules</td>
<td>● Additional education necessary</td>
</tr>
<tr>
<td>● Explicit business logic</td>
<td>● Additional task to be fulfilled</td>
</tr>
<tr>
<td>● Business in charge of adaptation of</td>
<td>● Remaining ambiguities because of</td>
</tr>
<tr>
<td>applications’ behaviour</td>
<td>different business worker</td>
</tr>
<tr>
<td>● Applications are not barrier for change</td>
<td>● Simpler exchangeability of business worker</td>
</tr>
<tr>
<td>● Easy integration of new laws</td>
<td></td>
</tr>
<tr>
<td>● Alleviation of mergers and acquisitions</td>
<td></td>
</tr>
<tr>
<td>● Improved individual customer support</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.** Advantages and Disadvantages for business’s worker and adaptability

6 Conclusion

The business rule approach seems to be a promising approach to remove implicit business logic caved within the different applications and to explicitly define the business logic with the help of business rules. This in turn increases the
adaptability of the applications’ behaviour according to the changing operations and decisions of the business.

Nevertheless, the business rule approach simplifies the effort needed in order to define and maintain explicit business rules to a great extent. Usually several hundreds to thousands of interdependent business rules might exist within one BRMS [18]. This leads to a complexity that is nearly not manageable and maintainable. Besides, it has to be kept in mind that business rules only express the business logic differently, but cannot reduce the complexity of the business logic that is found within the business.

References