Exceptions are just ‘some more rules’

Many times I have visited organizations that have told me “things got so complicated because of the many exceptions in our organization.” After an analysis of what is really going on I often find that they have an IT system that lacks support for handling many of the cases. The IT system and supporting processes were once designed to handle 80% of the cases automatically. But now the reality is that the IT system handles only 40% to 50% of the cases automatically and the organization is overworked handling all the ‘exceptions’.

I don’t want to claim that I have some miraculous recipe for getting better performance for this IT department. I do believe, however, that describing the knowledge and know-how of the people who work on the exceptions will help the IT department. The business rules approach and appropriate guidelines (for example *RuleSpeak*™) can help to describe all the decisions that are addressed in handling exceptions. But how do we describe the exceptions themselves?

In this column we will have a closer look at what it takes to describe exceptions as business rules and we will see that the result is just another set of consistent business rules. Give this set of complete and consistent set of business rules to your IT department, and I am sure it will be easier for them to increase the amount of automated support.

What is so complicated about handling exceptions?

Making sure you are complete (without the use of ‘else’)

Let’s take the example of a library where the process of applying for a library card is automated. The library card is differently priced for different audiences. Based on an interview with someone working in the library we get the following transcript:

> *“For example, students get the library card for free, elderly get a reduction, and children get a reduction as well unless their parents have a library card; in that case, they can get a library card for free if the parents upgrade to a family card.”*

So we have different prices for students, families, children, and seniors. Are we complete? No, we are not complete because we missed the person without children, who is neither studying nor old!

When we have many different rules, handling different situations, we need to think about all the different variations in advance so that we can check if we are complete.

Making sure you are consistent

Let's continue this library example and describe some more of the library's rules. The general rule is that you pay $20 for a library card. In RuleSpeak this can be described as:

- The price of a library card must be $20.

> *Students get the card for free, so that results in the rule:*

- The price of a library card for a student must be $0.

Oops, what we have created now is a direct conflict between the first rule and the second rule. There is no indication in the first rule that it should not be applied for students!

What options do we have? How can we solve or avoid this situation?

How should we handle exceptions?

Use a conflict resolution strategy

One way to deal with exceptions is to solve the conflict using a conflict resolution strategy. There are many well-known strategies to choose from (and, if you want, you can even describe those strategies as rules).

One very common strategy that has its source in the IT (expert systems) community is to prioritize the rules and to apply the one with the highest priority. Only if that rule does not

produce a result are the rules with lower priority (in that order) used. In our example, we would give the *student's rule* a higher priority than the general rule to make sure that it is applied first.

Another conflict resolution strategy is inspired by the legal domain. Here the most specific rule prevails over a more general rule. This strategy is used in court when two laws can both be applied. In our example, the *student's rule* is more specific to the case of pricing a library card so that rule would prevail over the more general pricing rule.

Both these strategies are applied in real world situations (IT and legal reasoning). Should we use them as a strategy in the business rules domain?

The answer is 'no'. We should not be satisfied with a conflict resolution strategy. Instead, the only good strategy is to solve the complexity. The reason is that without addressing the conflict we violate two of the three fundamental principles of SBVR:

- ***Accommodation Principle:*** *An element of guidance* (i.e., rule) *whose meaning conflicts with some other element(s) of guidance must be taken that way; if no conflict is intended, the element(s) of guidance must be expressed in such a way as to avoid the conflict.*
- ***Wholeness Principle:*** *An element of guidance* (i.e., rule) *means only exactly what it says, so it must say everything it means.*

Without these principles, action cannot be taken based on the rules because you never know, after reading the rule, if there is another rule that applies to the given situation.

Solve conflicts

What we should do is change the rule so that it applies only to the situation where it should be applied. We should do this in such a way that the list of conditions to be met is the most explicit. Preferably we use lists of conditions to make them as explicit as possible. For example,

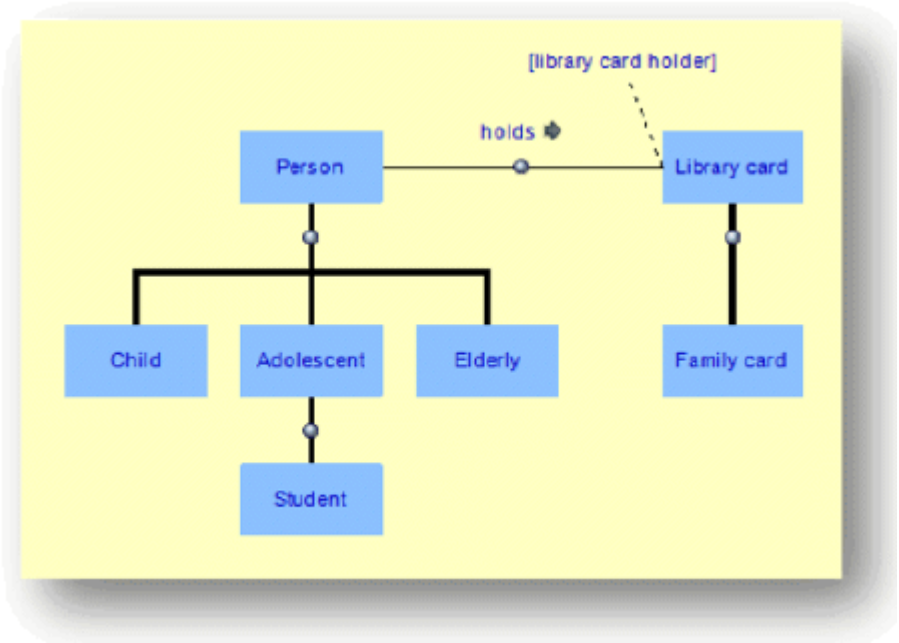
> **The price of a library card must be equal to $0**
> if all of the following conditions are met:
> • **The card-holder is a student.**
> • **The card-holder is not a family member of a person that holds a family card.**

The price of a library card must be equal to $15
if all of the following conditions are met:
• The card-holder is an elderly person.
• The card-holder is not a family member of a person that holds a family card.

The price of a library card must be equal to $5
if all of the following conditions are met:
• The card-holder is a child.
• The card-holder is a family member of a person that holds a family card.

The price of a library card must be equal to $20
if all of the following conditions are met:
• The card-holder is not a child.
• The card-holder is not an elderly person.
• The card-holder is not a student.
• The card-holder is not a family member of a person that holds a family card.

These rules should be read with this fact model in mind:



Alternatively, instead of this rather lengthy list of rule statements, we can use a decision table.

That also provides us with a better overview of our coverage of all situations:

		relation kind			
		elderly	student	child	none
family member of a family card holder	yes	*15*	*0*	5	20
	no	15	0	*20*	20

(Note: The cells marked in yellow were not explicitly mentioned in the rule statements.)

Create explicit rules that state how to handle an exception

In SBVR the assumption is made that “anything that is not expressly prohibited is assumed permitted, and anything not expressly declared as impossible is assumed possible.” We call this the *light world assumption*.

But at times there may be a need for a *dark world*: “anything that is not expressly permitted is assumed prohibited, and anything not expressly declared as possible is assumed impossible.” Especially for access to resources that are deemed sensitive, dangerous, scarce, and/or valuable in that situation, it makes sense to make a blanket prohibition and then selectively grant permission — this is called *authorization*.

The following example illustrates how we can define rules that make the ‘act of making exceptions’ explicit. The approach first defines the general rule. This general rule declares that some given area of business activity is prohibited except where there is some explicit ‘advice of permission’ given. For example, we would state that:

> *A person may borrow a book from the library only if the person is authorized by the library for book borrowing.*

This rule declares the general rule (no books may be borrowed unless the person is authorized). Now we can declare the specific situations in which book borrowing is authorized:

- A library card holder is authorized by the library for book borrowing.

And another one (for the sake of the example):

- A social security card holder is authorized by the library for book borrowing.

Without the explicit permission granted by these rules, there would otherwise be no authorization for book borrowing.

Now that our exceptional cases are described by ‘just some more rules’ it is easy to see that they can be supported by software systems.

Next time …

In our next column we will take a look at the options the IT side has for handling exceptions.

This article was originally published by BRCommunity (link).