

Robert Singer and Stefan Raß

---

## Abstract

The subject-oriented approach to model and execute business processes can be conceptually and easily combined with the business rules approach. Business rules are a mean to enhance the agility of workflows, as it should make the knowledge and decisions of an organization more explicit. As with the process model, ordinary users should be able to create, change and maintain the sets of rules in an idealized scenario. We demonstrate a real case for the beneficial use of business rules in the case of process automation and show a practical integration with an S-BPM reference implementation. Based on experiences in the field, we also point out that there is still serious lack of knowledge about actual trends and technologies in the context of the digital transformation of a business. The contribution is intended for practitioners with some interest in IT support for business process management.

---

## 11.1 A Business Rules Primer

Business rules and business processes are concepts which are closely connected. Nevertheless, we see the domain of business process management (BPM) as a general concept to manage a firm and business rules (BR) as a *refinement* to make BPM a more agile approach—especially in the case of the automation of (parts of)

---

R. Singer (✉)

FH-Joanneum—University of Applied Sciences, Alte Poststraße 147, 8020 Graz, Austria  
e-mail: robert.singer@fh-joanneum.at

S. Raß

StrICT Solutions GmbH, Plüddemanngasse 39, 8010 Graz, Austria  
e-mail: rass@strict-solutions.com

processes. Business rules are—as will be discussed soon—a tool to give process participants more and easier control over their business processes. Business rules are also a means to collect and codify parts of the tangible knowledge of a firm or organization.

### 11.1.1 Introduction

All organizations operate in accordance with a set of underlying principles—that is what “organization” means. These principles define the “business logic” which controls the way the business conducts itself. Basically, a business rule (BR) is a compact statement about an aspect of a business and can be expressed in terms that can be directly related to the business. Business rules use simple and unambiguous language that’s directly accessible to all interested parties in the organization.

Rules do not only articulate some constraints, but also provide a means of encapsulating knowledge about the business. Rules cannot stand in isolation but need to be rooted in a rich representation that captures the overall facets of a business. As rules are constraints, they define conditions that have to hold true in specified situations. Business rules and processes can be beneficially integrated for a more agile setting of models, defining how to conduct certain aspects of a business. In concrete, business rules can be integrated into a business process model in such a way that at a decision point in the model there is a reference to a concrete rule, the rule is evaluated and depending on the result (e.g., true or false) the process continues. For example, a rule could define the following logic: all purchasing orders exceeding a total amount of EUR 5.000 need to be confirmed by the department manager.

Business rules can be expressed using different levels (informal, formal) of expression as illustrated with the following examples:

```
The weight of a parcel has to be below or equal 32kg.  
parcel.weight <= 32
```

One additional remark: it should be clear that such numbers (the 32 kg in the example above) should be handled as *business parameter* in the same way as rules; that means they are stored together with the rules and do not appear as *magic numbers* somewhere hard-coded in the source code. So, if the rule still applies, but the value changes, anybody can easily change the value, and from that moment on all business process instances with a reference to this rule will use it—voilà. No software developer is needed (as long as the front end to manage the rules follows common guidelines for usability).

### 11.1.2 Illustrative Example

*Microsoft Outlook* is a well-known desktop program to manage e-mail messages, appointments, contacts, and other facets of business life. The program integrates

with other applications, such as *Microsoft Exchange*, to provide a range of facilities. Here—following the idea of Morgan (2002)—we are going to have a short look how *Outlook* uses rules to automate the handling of mail messages.

Rules in Outlook are composed by using a “wizard” that provides an interactive dialogue from which you can define various parts of your rule. The interface design shows how a potentially complex and technical task can be made much easier for ordinary users ... (??)

*Outlook* is not a fully functional business rules system, but it demonstrates the core concepts and even is an example of compromises that may be found in your own applications.

Each rule statement is made up of a number of logically connected clauses. The first clause defines whether the rule applies to outgoing or incoming messages. There can be zero or more condition clauses which all together have to evaluate to true to activate the rule. At least one action clause has to be defined; additionally, zero or more exception clauses have to be false, or the rule will not be activated. A typical example for an *Outlook* rule might be

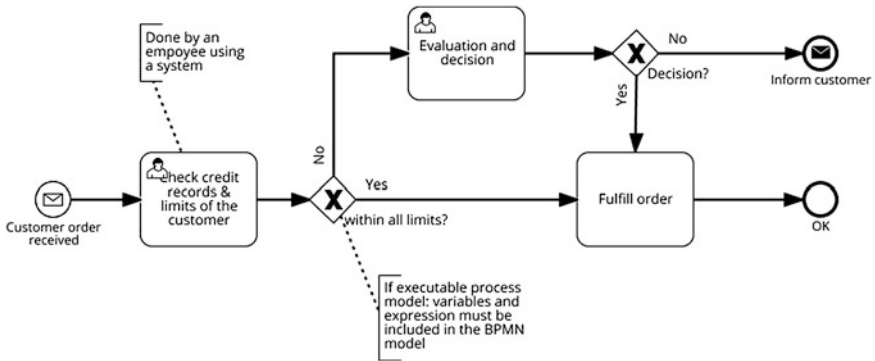
```
Apply this rule after the message arrives
where my name is in the To or Cc field
  and which has an attachment
move it to the folder attachments
  except if sent only to me
```

All possible conditions, exceptions, and actions are chosen from a predefined list. You can also use other mail clients to define your rules; rules can be client side (executed on your computer) or server side (executed on the mail server). It is clear that this way of defining rules needs to be implemented in the software application; there is code to execute the defined rules based on a predefined set of logical constructs.

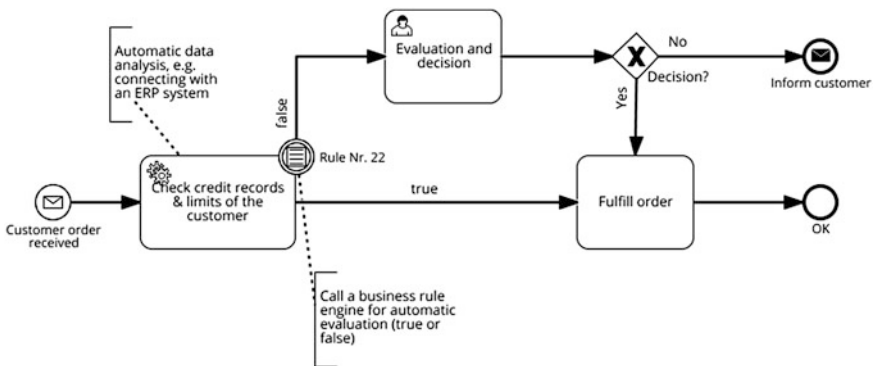
### 11.1.3 Business Processes and Rules

Principally, integrating business rules with business processes is as illustrated in the *Outlook* case. What we need is a way for ordinary users to build rules based on predefined sets of clauses and parameters. The business rules concept should hide the implementation aspects from the users, giving them the possibility to define and change the behavior of a business process at defined decision points without coding. Nevertheless, the concept of business rules is a general purpose tool for software developers to generate more agile applications.

We demonstrate the concept with the help of a small process as depicted in Figs. 11.1 and 11.2. We intentionally use BPMN to emphasize that the integration of business rules and process is a general concept; in the next section we will discuss the integration using the S-BPM methodology as a foundation for *enforceable* business processes.



**Fig. 11.1** Simple process model with a decision point. Technically, at the decision point the value of a variable is evaluated and depending on this value one of two alternative process paths is executed. The value is set in the preceding activity by a human agent



**Fig. 11.2** The same process as in Fig. 11.1, but using a business rule for automation. After receiving a new order the data is analyzed and rule number 22 is called—using a business rules engine. The engine evaluates the rule and returns a value (true or false in this case) back to the instance of the process. Based on this result the process engine can decide which of the two alternative paths to choose. The rule itself is stored in a central repository and can be modified independently of the process model

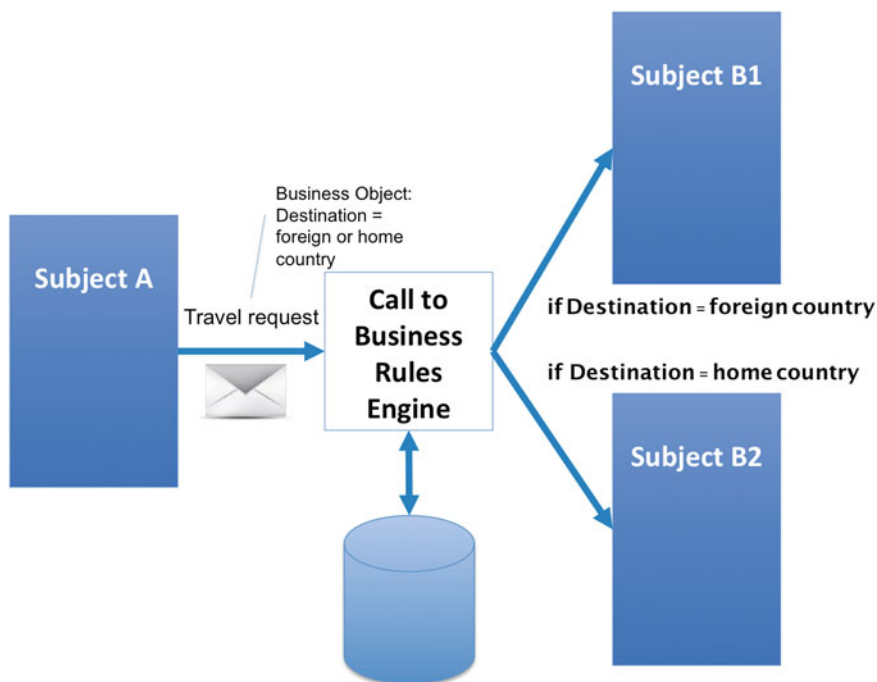
There are several books about the topic available. The standard book and valuable first reader is *Principles of the Business Rules Approach* by Ronald (2003). The book *Agile Unternehmen durch Business Rules* by Schacher and Grassle (2006) gives a very good overview, but as the title shows, it is in German; the technical part is somewhat outdated.

## 11.2 S-BPM and Business Rules

### 11.2.1 Concept

How can we now integrate the business rules approach with the S-BPM methodology? Interestingly, there are two possibilities for how to integrate calls to a business rules engine:

As there are messages sent between subjects, we could define business rules to evaluate logical expressions based on business objects included in the messages. That means we could modify the message and or the content of a message (the business object) based on a set of rules; for example, we could *automatically* modify the receiving subject(s) depending on evaluations of the containing business object. So, if the business object holds information about a customer order we could change the *subject* depending on the amount of the order. Technically, there have to be some *locations* in the application from where we can call business rules (the business rules engine) (see Fig. 11.3 for illustration). A distributed environment can be established using, for example, web- or other service-oriented technologies. Depending on the process model we can think of local and global business rules. So we could define business rules valid for message exchange within an organization



**Fig. 11.3** If all messages would be routed over a central “rule box”, depending on the message type and or business object or business object content one or more business rules could be applied

and with other organizations. This can be an interesting application for compliance or other topics.

Another possibility is to call business rules from function states defined in the internal behavior of a subject, i.e., using the concept of *refinements*, an integral part of the S-BPM methodology. In general, a *Refinement* is some extra code which can be invoked by a running instance. Depending on the result of the automatic evaluation of the rule, a decision can be made. One fact has not been mentioned yet: in many situations there is a decision tree built of more than one rule; this can be a chain of IF ... THEN ... ELSE ... clauses. The last ELSE could be used to initiate a human-based decision.

The elegance and flexibility of the S-BPM methodology can be seen once we recognize that an extra component is not needed to evaluate rules on the messages exchanged between subjects. So, we could define a subject *with the internal behavior of evaluating rules on incoming messages*. It is an integral part of the S-BPM methodology that a subject be a conceptualization of an agent—and an agent can be human, a physical machine (including an electronic interface for communication) or a software agent. Hence a software agent could be “intelligent”, if we want to use a common term for agent systems. Hereby, we mean agents reacting on input of the environment and making autonomous decisions (but only based on internal behavior defined using an algorithm).

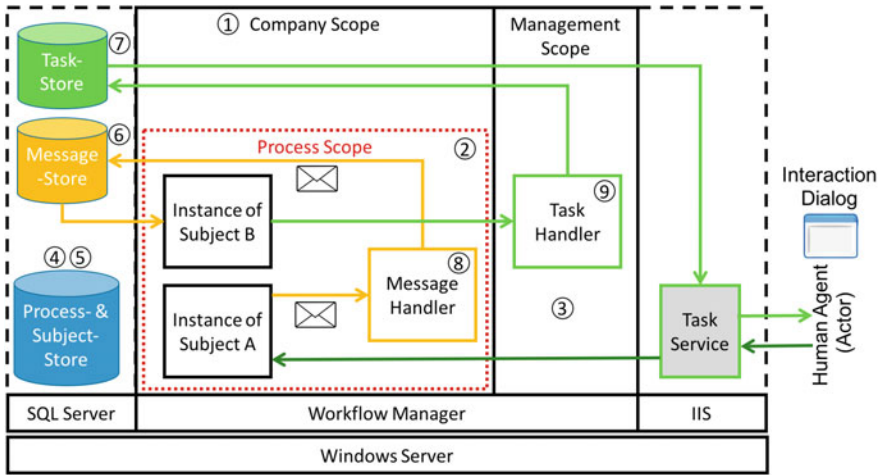
In many cases there is not only one rule; the application of certain rules can be summarized in *decision tables*, as depicted in Fig. 11.4. Decision tables are easy to understand (but can be large).

### 11.2.2 Implementation

The core idea of business rules is automation, or as we call it today “digital transformation”; therefore we need technology to collect, store, manage and evaluate rules. A step towards standardization has been done by the *Object Management Group* (OMG) defining the standards *Semantics of Business Vocabulary and Rules* (SBVR) and *Decision Model and Notation* (DMN). Nevertheless, there is

Conditions	Rules					
	1	2	3	4	5	6
C1. Parcel weight <WeightLimit	Y		Y	Y		Y
C2. Parcel size <SizeLimit		Y	Y		Y	Y
C3. Urgent Delivery			Y			Y
C4. Region 2	Y	Y	Y			
C5. Region 1				Y	Y	Y
Actions	1	2	3	4	5	6
A1. Use Service Provider 1	X	X				
A2. Use Service Provider 2			X			X
A3. Use Service Provider 3				X	X	

**Fig. 11.4** A decision table is a matrix of rules and actions. For any possible combination of conditions, actions are defined



**Fig. 11.5** StrICT architecture. The processes are executed server side and the workflows are coordinated through message exchange (orange). Task requests (light green) and task answers (dark green) are routed to a client via the task service

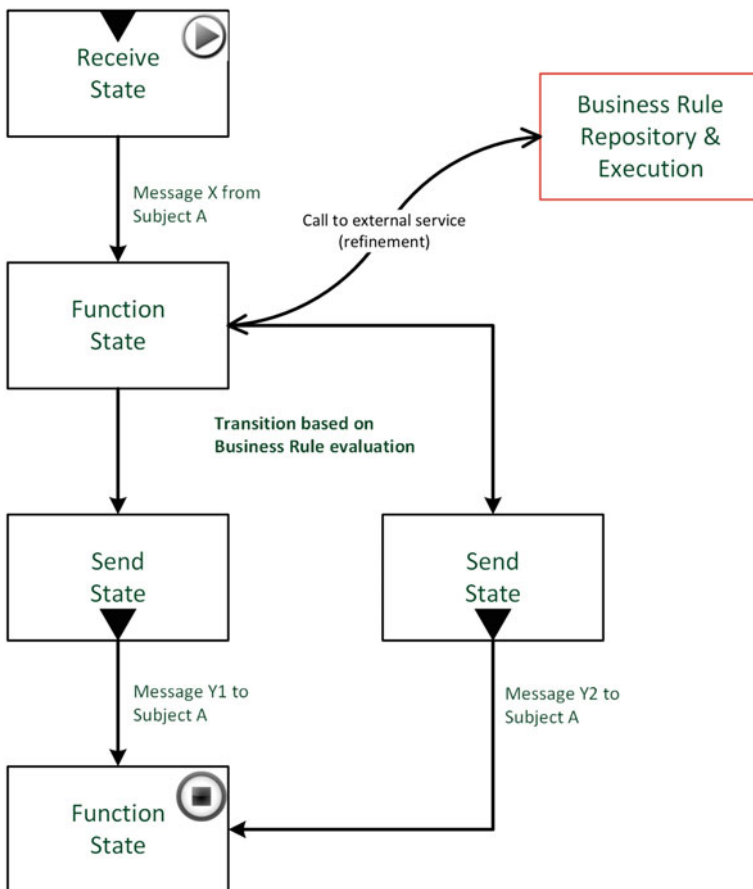
no one standard technology (implementation), but there are several—mainly commercial solutions—available.

We have integrated the business rules approach with the *StrICT*<sup>1</sup> S-BPM implementation (Singer et al. 2014; Singer and Raß 2015). The architecture is depicted in Fig. 11.5. Processes are hosted on an instance of the *Workflow Manager* (WFM), which is responsible for the hosting, administration and configuration of the subjects based on scopes, such as a *Company Scope* (1) for the processes of one organization, a *Process Scope* (2) for each process and a *Management Scope* (3). Each company has its own *Process Store* (4) and *Subject Store* (5); the same for *Message Store* (6) and *Task Store* (7). Each company has *Task Handler* (9) instances to generate new tasks and each process has *Message Handler* (8) instances to manage message exchange. *Task* and *Message Handler* are implemented as workflows. The mechanism of *Scopes* ensures full encapsulation of one company or organization by the other. Further, it allows permission management on a very fine granular basis for each activity; depending on the rights of a role, activities can be seen or not, executed or not.

The architecture of the StrICT Windows reference implementation is depicted in Fig. 11.5, as shortly discussed above. All messages of an agent are handled by a message handler (denoted by “8” in the figure). Within this handler we could implement a call to a business rules repository and an execution engine for evaluation. At this point the message itself and or the content, i.e., the business object, could be modified.

<sup>1</sup>StrICT = Structured Information and Communication Technology.

Such a mechanism would mean change to the architecture itself, so we decided in a first attempt to implement the use of business rules in function states of the internal behavior of subjects (see Fig. 11.6). This offers the flexibility—as discussed beforehand—to define so-called “intelligent” subjects. Subjects use rules defined in a business rules repository to *refine* their behavior, for example to make decisions based on rules or a set of rules. In this way we can define—as usually when using the S-BPM paradigm—human interaction processes or intelligent agents without human interaction. Technically, any subject is defined as a workflow, based on the *Microsoft Windows Workflow Technology* functionality which is implemented in the operating system routines (to be more exact: in the relevant .NET libraries) and the *Windows Workflow Manager* server component.



**Fig. 11.6** A conceptual example of the internal behavior of a subject (SID). From any *Function State* we are enabled to make a call to a business rule system; depending on the result it is possible for us to make a decision



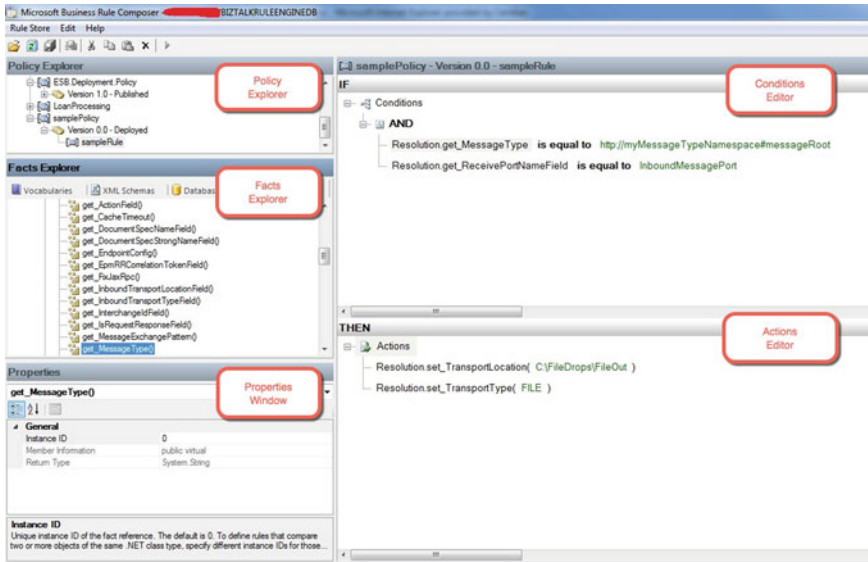


Fig. 11.7 The BizTalk business rule composer with sample example data for illustration purposes

Now, to enhance the actual architecture we can refer to the functionality of the *BizTalk* server component. This includes a business rules composer (see Fig. 11.7) to design rules, which are then stored in a database instance (SQL server instance). Rules can be evaluated, too. Additionally, the *BizTalk* server integrates with the *Microsoft Service Bus* component, an important aspect of the used reference implementation. An obvious disadvantage using the *BizTalk* composer is its poor usability; the composer needs a lot of technical skills and therefore we could not meet one requirement, namely the involvement of ordinary business users in designing rules. Nevertheless, that is one of the typical problems we experience in the field; not all problems can be solved immediately.

### 11.3 The Case of ABC Logistics

Now, let's look at a real-world project which addresses the discussed concepts. The project uses the discussed technology platform to execute S-BPM process models, including calls to a business rules execution engine (a technical prototype for further experiments):

The company *ABC Logistics*<sup>2</sup> offers the service of sending parcels all over the world. Towards customers they present themselves as a globally acting logistics provider collecting orders by web site (structured information) or e-mail

<sup>2</sup>The case is real, the company name has been anonymized.

(unstructured information). In reality the company is an intermediary using third party services to fulfill customer orders. Nevertheless, the company has its own facility for collecting, sorting and labeling of the parcels. There are no predefined limits in size or weight of a parcel; the core competence and offered value proposition (mainly towards business-to-business customers) is the knowledge about which logistics provider to choose in a specific setting.

Sending a pallet to Saudi Arabia? Sending a small and very urgent box to Shenzhen? Sending a medium box with breakable content to Kenya? *ABC Logistics* knows the best (in a holistic view) provider to contract—invisible for their customers. Such decisions are done manually during all stages of the core business process. This can be done only in an error-prone and costly way. Moreover, a recently installed automated packaging line cannot be used in the intended way, as there is no coherent and persistent data model available over the whole process chain—so, all data has always to be interpreted by human actors. Consider for example the worst case: a customer sends information for a parcel via e-mail; then somebody has to extract manually all the needed data from the mail, check the data, maybe contact the customer if information is missing or unclear, check size and weight of the parcel, contact the customer in case of serious differences between customer and real measures, decide which contractor to use for delivery, enter the data into the system of the chosen contractor, print out any documents and labels, and label the parcel. Currently there is a pool of about 15 service providers, all using their own IT systems.

The main tasks for a digitized business process for *ABC Logistics* now are:

- a single coherent data model, i.e., business object for the whole process
- as much as possible automated decision making to determine the optimal (cost and quality) service provider
- fully electronic integration of the packing line or other physical devices with the digitized process
- integration with the company web site (order management and customer service)
- integration with an ERP system, including bookkeeping

What are the promises of a fully digitized business process for *ABC Logistics*?

- mostly automated decisions about which service provider to use in a concrete transaction based on codified rules; the core idea is that the rules are readable and changeable by ordinary process participants
- efficient use of the investment in the automated packaging line
- reduction of errors based on an integrated information model
- reduction of transaction costs
- gained efficiency enabling growth, i.e., an increase of rate of transactions based on a decrease of throughput time
- increase of efficiency and reduction of errors based on the integration with other systems

In the following section we will discuss the first steps and findings towards a solution for *ABC Logistics* based on S-BPM, which conveniently offers all needed capabilities to digitize ABC's business processes.

---

## 11.4 Results

### 11.4.1 Impact of Actions

Now, what is the focus of *ABC Logistics* in this case? Interestingly, it is the focus on automation of the data transfer (the interface) to their service suppliers. They do not realize that an automatic data transfer is useless without a digitized business process.

Consequently, the first step in the project in the field is to study (technically) how to automatically exchange data with the logistics partner. At the moment most of any interaction with partners is done manually via web sites; in the worst case a remote connection and data entry on the remote server needs to be established. This is rather inefficient, but accepted by *ABC Logistics*. For an ideal solution of a human interaction workflow, in this case with as much automated decision-making and data exchange as possible, it is clear that this is a necessary part of the solution.

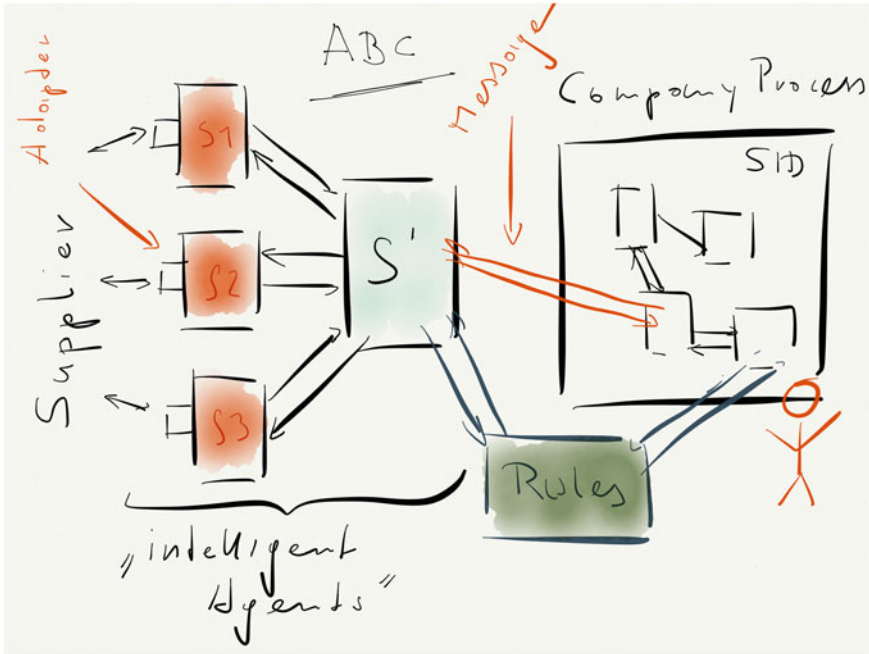
But this focus does not consider the following topics:

- Data collection and consolidation still has to be done manually
- Data exchange with suppliers and customers still have to be done manually
- The packaging line is still not integrated in the process

The reason of this managerial priority setting is the fact *that a well-designed and semi-automated business process is not seen as an asset, but more or less as nice to have*. Some sophisticated data interface, i.e., information technology, is seen as the benefit. That is somewhat surprising when reading all the books, papers, reports and blogs about business process management—however, that is the reality in the field.

So again we have an IT project without clear requirements and unclear business processes, and we have the feeling that it is of no good. What can be the conclusion? It still seems that there is a lack of understanding of the topic of business process management in the field—especially in the case of an integrated view with technology. No operational business process nowadays can function efficiently without support of technology.

The concept of our proposed solution could be as depicted in Fig. 11.8. In the beginning there should be a focus on the core processes (as they are now) to understand the information flow. This includes the design of the required business object, which results in a coherent data model for the process. In the first shot there will be no integration of external systems. Concurrently, there can be a start to the development of intelligent agents (a service composition) to handle messages from



**Fig. 11.8** A sketch on the napkin for a possible solution scenario for *ABC Logistics*. First, the end-to-end process of the company has to be modeled; afterwards or concurrently a set of business rules needs to be defined to model *decisions* in the business process. A technical challenge lies in the development of the *interfaces* toward the customer systems (*left side boxes*). The “S” denoted *box* represents an *intelligent Agent* (subject), the working horse for automated decisions based on the defined business rules set

other (human) agents to automatically transfer data from and to service suppliers. Later on, this network of agents can gradually increase its “intelligence” when more and more rules (knowledge) are incorporated in their internal behavior descriptions.

### 11.4.2 Open Issues

Nevertheless, to come back to the integration of business rules with subject-oriented business processes, it does not seem to be a great effort to integrate both concepts technically; however, there is still a lot to do to provide much software enabling ordinary process workers and responsible managers with convenient user interfaces. There also seems to be a real need in the field for a **better education of practitioners**, especially in small and medium enterprises (SME), as they usually do not have specialized staff for business process integration. It definitely is not sufficient for a CIO to be able to configure a mail or database server. All surveys of the last years stress the fact that IT departments also should focus on business and business

process management—that includes deep knowledge and understanding of business process execution.

Now, some final technical remarks about business rules integration. Not considered yet in the reported case are topics, for example, such as *forward* and *backward chaining* of rules (Morgan 2002). There are also some more possibilities to collect and store business rules. Additionally, there are also several business rule execution engines available (open source and commercial ones). Another point not evaluated is performance; but we can expect a commercial product such as *BizTalk* server to be capable of evaluating a huge number of rules concurrently. These and other topics have to be evaluated in further projects.

### 11.4.3 Takeaway

Seen from a purely technical point of view the integration of business processes and business rules is ready for application. S-BPM is a perfect candidate for achieving that, as the concept of rule evaluation is inherently included in the concept of message exchange—a subject sends a rule (question) to a rule evaluation subject and receives the evaluated rule (answer = decision). This mechanism enhances the agility of business processes, as it should drastically reduce the effort to adapt process models. *Actors* should be given the tools to change process models more often without support of *Experts*. As demonstrated, there are many useful application scenarios in real-world business processes, especially also in the field of logistics and manufacturing—notably if we develop integration scenarios as discussed in the context of *cyber physical systems*.

**Open Access** This chapter is distributed under the terms of the Creative Commons Attribution Noncommercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

---

## References

- Morgan T (2002) Business rules and information systems. Addison-Wesley, Boston
- Ross Ronald G (2003) Principles of the business rule approach. Addison-Wesley, Boston
- Schacher M, Grässle P (2006) Agile Unternehmen durch Business Rules: Der Business Rules Ansatz. Springer, Berlin (in German)
- Singer R, Raß S (2015) Structured communication—approaching S-BPM with Microsoft technologies. In: Fleischmann A et al (eds) S-BPM in the wild. Springer, Berlin
- Singer R, Kotremba J, Raß S (2014) Modeling and execution of multienterprise business processes. In: 16th IEEE conference on business informatics, workshop on cross-organizational and cross-company BPM (XOC-BPM). Genf, Switzerland, 14–17 July