visual rules®

# - White Paper -

# Business Rules and SOA: Parallels and Synergies

Innovations Softwaretechnologie GmbH
Ziegelei 7 - 88090 Immenstaad
Germany
Tel. ++49 (7545) 202-500
www.visual-rules.com
info@visual-rules.de

Last revised: January 2006

# Table of contents

# 1. Introduction

Both Service Oriented Architectures (SOA) and Business Rules Systems (BRS) aim to adapt business processes quickly and easily to new requirements (e.g. changes in the law, time to market, mergers & acquisitions).

The SOA approach offers the concept of services for this. Services are functional units for clearly defined areas of expertise. They are defined from a functional view and via service brokers can be used company-wide and beyond.

The Business Rules approach includes the concept of encapsulation. Business rules are developed separately from the technical environment. This allows business logic to be quickly adapted to changing requirements. The goal is to make business rules available company-wide.

SOA and BRS complement each other in the sense that both disciplines are driven by functional requirements and are used company-wide. However, there is a clear separation of duties between SOA and BRS. SOA delivers the mechanisms for interaction between applications; BRS handle the encapsulation of business logic in services (rules in services) or develop complete rule based services (rule services).

## 2. The Business Rules approach

Experience in the use of BRS shows that little regard was paid to the encapsulation of business logic in the architecture of applications in use today, irrespective of whether these are host systems or server based systems. Business rules are for the most part closely interwoven with the technical parts of the program. Seemingly minor modifications to business logic often precipitate significant program modifications. Exacerbating the problem is that if program documentation is available at all it is useless for understanding the business logic implemented due to the mixture of functional and technical aspects.

Business Rules Systems solve this problem. With these business logic is structured and encapsulated in the application to make it easy to change (*see Fig. 1*).

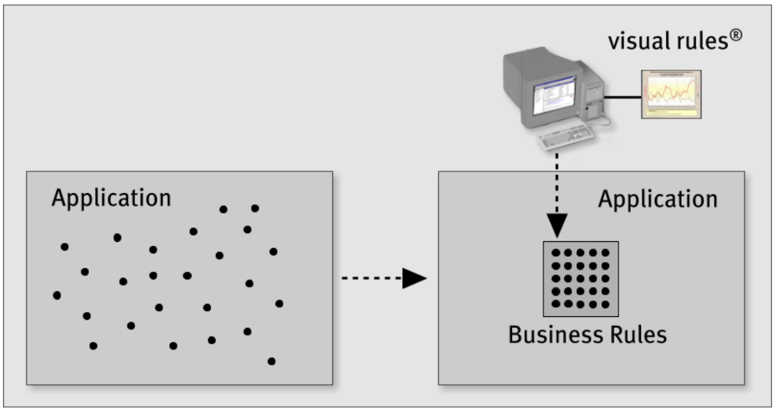

**Fig. 1**: Business rules are spread across the application on the left. In the application on the right they are encapsulated; development and monitoring is done using visual rules®.

The graphical modeling approach of visual rules® results in the mapping of business logic that can be easily understood and maintained. More than that: the business logic encapsulated this way can be tested separately and provides functionally motivated runtime information and execution logs. visual rules® also handles the automatic documentation of business logic. The BRS thus supports the entire development cycle of business logic.

# 3. Interaction as a mission critical factor

In the real world you are not dealing with individual applications, but with a mature application landscape. Continuous process optimization demands that applications that were clearly separated in the past must increasingly interact across platforms on the business logic layer.

The quickest solution for interaction between two applications is establishing a point-to-point connection (*see Fig. 2*). This solution is tailored to the applications involved and therefore not coordinated company-wide. Such solutions work in the short term. However, the longer development is done like this, the more application-specific interfaces arise. This increases dependencies and the overall system becomes unmanageable. The resulting technical and functional network of applications becomes increasingly resistant to change. The effects of selective changes are no longer foreseeable.

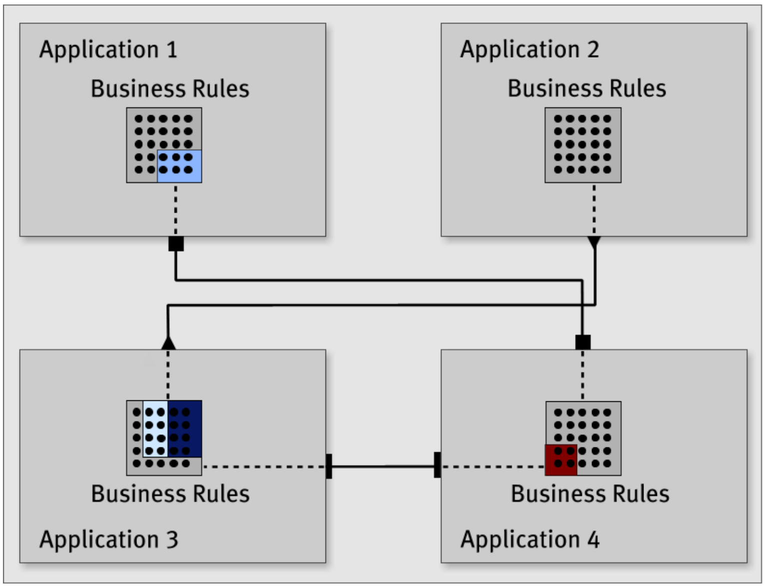

**Fig. 2**: The interaction between applications is point-to-point; the interfaces are application-specific. Changes can only be made with difficulty.

The mission critical status is reached when market requirements can no longer be sufficiently or quickly implemented and the company is more hampered than supported by the IT system.

# 4. EAI and BRS

One way out is offered by Enterprise Application Integration (EAI) approaches whose goal is to provide an unrestricted communications platform for interactive applications within an organization. For this the hub and spoke approach (*see Fig. 3*) includes a central integration component in the form of a message broker (hub) for every application (spokes).

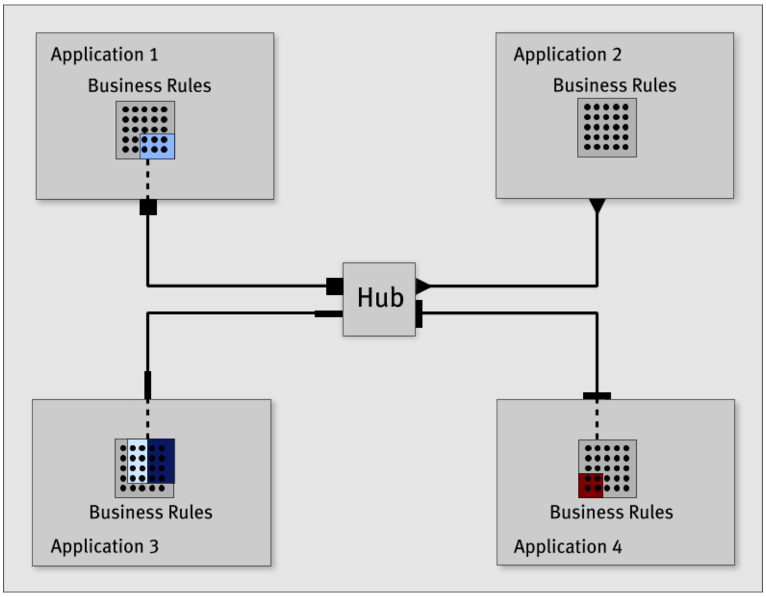

**Fig. 3**: The interaction between applications takes place through the central message broker (hub). Business logic can be shared among applications.

When an application accesses the business logic of another application, communication always takes place through the hub. The functional design and the connection of source and target applications are now independent from one another. The result is that the technical dependency between applications is resolved and inter-application use of business logic takes place. The architecture is clearly structured and can be expanded dynamically.

One disadvantage of such an architecture is the higher than average time and effort required to adapt integration components, if new applications have to be integrated or old ones removed, because a uniform definition of the functional interfaces is missing – they are proprietary. The conceptual claim made by EAI, besides delivering technical and functional integration, therefore breaks down halfway, leaving this approach a purely technical solution.

# 5. SOA and BRS

## 5.1. Rules in services

The interaction between applications on the business rules layer is only really possible with SOA. This is because SOA provides the technical platform with the Enterprise Service Bus (ESB) and a standardized functional specification of services with the Web Services Definition Language (WSDL). The interaction of services – the "orchestration" – takes place by means of the Business Process Execution Language (BPEL).

SOA provides interaction on different layers (*see Fig. 4*). To do this, an application is broken down by functional aspects into workflow services, business services, component services and data services (layering). The services on different layers each have their own specific business logic that is also provided to other services via the SOA. SOA and BRS thus ideally complement one another by making the BRS an integral component of an SOA.

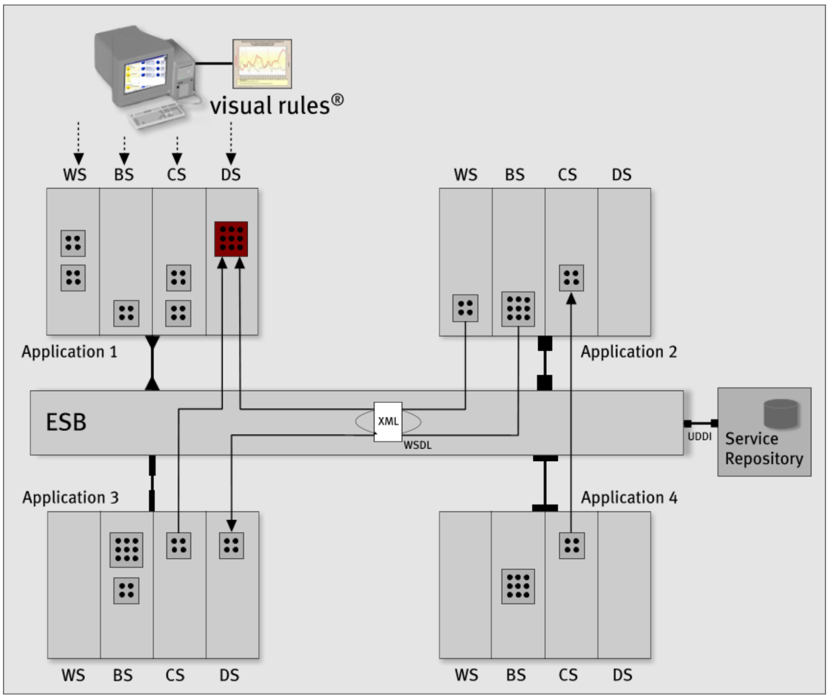

**Fig. 4**: The application was broken down by function into workflow (WS), business (BS), component (CS) and data services (DS). The services contain the encapsulated business rules developed with visual rules®.

This is where the different approaches of Business Rules Management Systems (BRMS) diverge. Classical providers of BRMS follow the approach of managing business rules in a central repository. However, a rule repository centralizes the business logic of different service owners – and thus stands in the way of building a flexible service architecture whose services are managed via a service repository.

visual rules® is positioned as a tool that supports the development of business logic in services and can be seamlessly integrated into the SOA. The visual rules® approach results in increased efficiency and standardization in the development of services. Even more, visual rules® implements additional features that enable monitoring of services business logic.

## 5.2. Rule services

Up to now we have looked as business logic as part of an application or service. The approach is relevant if the business logic of existing applications is to be provided as a service to other applications.

If however, new services are being developed with business logic or these are to be extracted from existing applications, the logic can also be provided using visual rules® as its own rule service (*see Fig. 5*). This is particularly wise if the service is used by multiple applications.

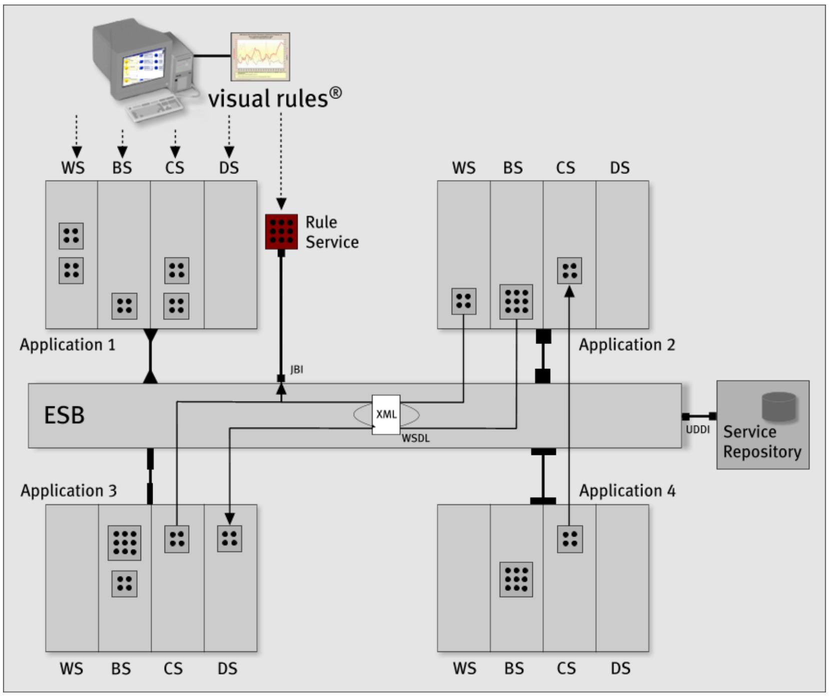

**Fig. 5**: Business rules in services and as standalone rule services; each are developed using visual rules® and their execution is monitored at runtime.

While the embedding of business logic in the “rules in services approach“ takes place via the frame application, for “rule services“ complete services must be generated that are capable of running independently in an SOA. To do this the BRS must not only be able to define business logic in the form of rules, but also implement the required interfaces so that rules can be integrated directly into the SOA without additional programming.

visual rules® makes it possible to model complex operations and generate complete Web Services (including WSDL and deployment descriptors, with visual rules® 3.3) from the models. Therefore, visual rules® is a qualified tool for the development of business rules in services and of its own rule services.

## 5.3. Decision services

Decision services are a special form of rule services within an SOA (*see Fig. 6*). They are deployed company-wide for the purpose of Business Activity Monitoring (BAM) in order to monitor business processing globally and to make decisions managed by rules.

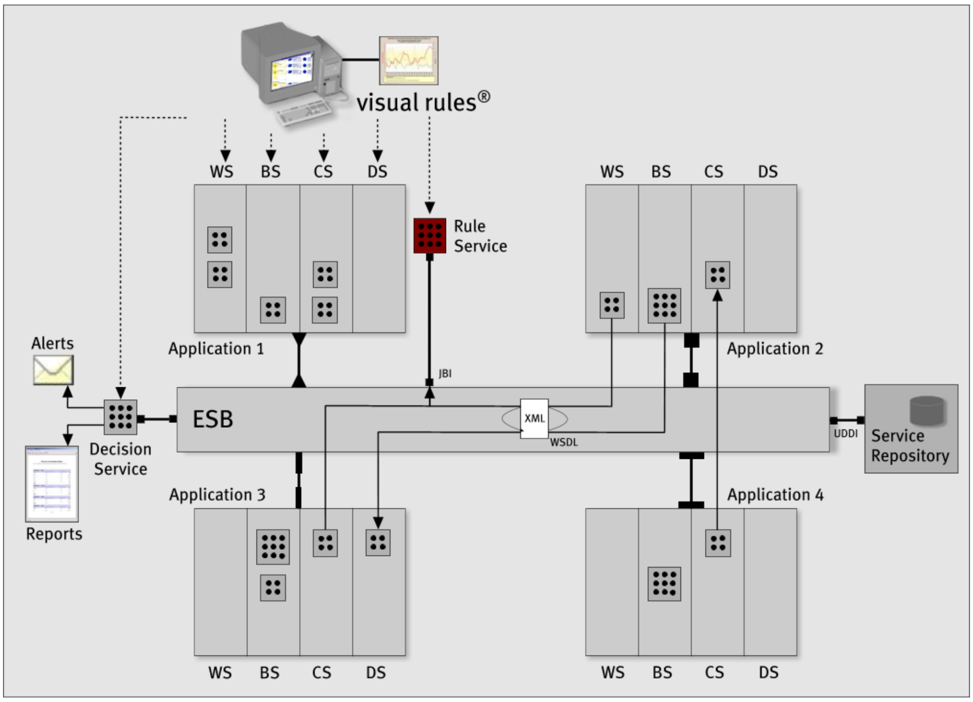

**Fig. 6**: The results of a decision service can be output as a BAM report or alert. The decision service is developed using visual rules®.

With decision services events are processed based on rules, interpreted, and defined actions are triggered. This happens when specific events occur, occur late or also fail to occur. The decision service then triggers the output of alerts or provides the information in a report. In this way rule-based BAM enables decision maker control at an early stage.

The generation of alerts or report information is controlled by actions in visual rules®. Actions in visual rules® are also extension points. You can thus create your own platform-specific actions in addition to the hardwired ones. visual rules® is used to generate not only the business logic of services, but also the logic for monitoring business processes.