

Design and Implementation of a Model for Business Rules Automation

Saqib Ali, Ben Soh and Torab Torabi

Department of Computer Science and Computer Engineering

La Trobe University

Bundoora, VIC, 3083, Australia

<mailto:{saqib.ali, b.soh, t.torabi}@latrobe.edu.au>

Fax: (61 3) 9479 3060

Abstract

In this paper we propose an approach to treat business logics and system logics separately. Using the proposed model, we can ensure that for any set of facts, business rules are applied efficiently and consistently.

In our model, it is also feasible to introduce new rules into any business process, where each process depends upon some specific rules, policies, procedures and standards. As a result, business processes can be automated in a more customizable, flexible, efficient and reusable way. In the paper we use a vehicle reservation system as an example to illustrate how our proposed approach can be implemented.

KEYWORDS: Design, Business Rules Model, Business Processes, XML, Component based development and reusable.

1. Introduction

Traditional application architectures are costly and difficult to customize. Only large enterprises application vendors can afford infrastructure for customisation, integration and workflow management. Successful businesses must have the flexibility to meet the varying needs of customers.

Large businesses in complex environments are made up of thousands of processes and their profit depends on efficient execution of the processes for delivery of goods and services. Automated business applications were initially introduced to handle routine processes, avoiding the more complex and rapidly changing areas of business rules and strategies. The traditional method of hard coding business rules into implementation has proved far too inflexible [4, 6]. There is a need to change how businesses can react and handle full range

of complexity and flexibility required for business environmental changes.

Most of business processes are directly dependent upon business rules. So there is a need to change the way business rules incorporated. In implementation of application also there is a need to make business rules more customisable and capable of providing help to business analyst during the development phase of business processes.

In this paper we propose a model to incorporate the use of business rules into business process development. We will present a background research in business system architecture and business model development. We will then develop a business system architecture that involves overall system architecture and business rules model.

In the following sections we will discuss the rule based system architecture. Then we will analyse business rules in process automation and present rule based platform approach. At the end we present an example with implementation and shows end results, which illustrates our approach.

2. Background: Business Rules

Business rules provide the knowledge behind every business structure or process. They are therefore at the core of functional requirement [2,5]. Business rules are based on policies and documents where these documents consist of procedures and standards that follow internal/external protocols. Business rules are the policies and constraints of the business. Business rules are a functional requirement, the decisions, guidelines and controls that are behind system functionality. We have explained in detail the sub parts of business rules model in our previous research paper [1].

We propose a new modelling approach that helps to visualise and analyse business processes at very beginning stage of system development. This provides a picture for business executives to see what needs to be done, by whom and when. This approach provides business-rules driven platform that not only creates a visual picture of the links between different events, but also creates a declarative, accurate, and actionable infrastructure for automation.

The solution is similar to three-tier business process architecture where user can interact with user interface, which is the first tier of the architecture. The second tier is process customisation, which is responsible for customisation of the business rules process according to business rule process expert's need.

At the last tier, business rules are implemented and incorporated into the business processes. Figure 1 represents a system with three-tier architecture.

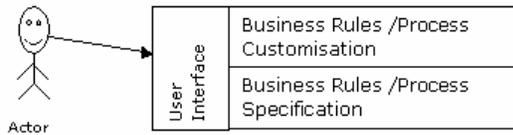


Figure 1: Three-tier system architecture

3. Motivation: Automation process for business rules model

Virtually all businesses specify business rules, whether as policies, or procedures. A business dictates how the organisation executes business decisions, processes, and constraints essential to the company's strategy [8,9].

Organisations automate their business processes with the aim of improving operational efficiency, reducing costs, improving quality of service offered to their customers, and reducing human errors.

There are many types of business rules. For example, a business rule might specify the constraints under which a preferred customer receives a benefit such as, "Mobile company gives their customers free upgrade from existing packages to new packages." Or a business rule might define attributes that must be validated, such as "Mobile dealers must be from Australia." A business rule might express process driven events, including notification procedures such

as "On upgrade of two packages or more from same customer must be notify the area manager."

There is a need to propose a model, which can enable a business to introduce their rules into any business process automation, where each process depends upon some rules, policies, procedures and standards. One of the objectives of the model is to make business process automation more customisable, flexible, efficient and reusable [1,3].

The modelling, development, testing and execution of business rules and workflow tasks need careful coordination to perform automation of business processes. By incorporating this approach, we can achieve error-free, efficient implementation according to customers' ever changing business environments.

4. Our Proposed Design

Our proposed business model design involves following steps:

- 1) Identify business components
- 2) Specification of each component
- 3) Identify relationship between each component.

A high level semantic representation of the business model has been shown in Figure 2 using a UML (Unified Modelling Language) diagram [15].

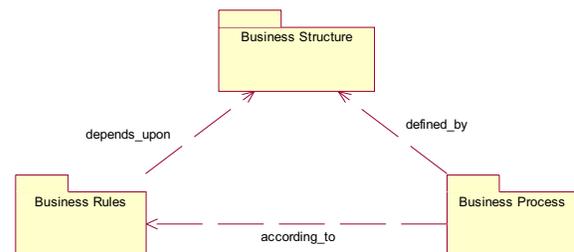


Figure 2: Components of business model

In this paper we present the overview of a semantic business model [16], which is composed of three separate components linked with each other, having specific relationships. Business structure is "defined_by" Business Process and Business Process is "according_to" Business Rules, where Business Rules has the "depends_upon" relationship with Business Structure [1, 17].

In the following UML diagram (Figure 3) we represents a model diagram of business rules, these business rules are a component of a business model, which we already explained in detail in our research paper [1]. In the paper, we only emphasise on the

business rules model component. (The business process component constitutes part of our future research.)

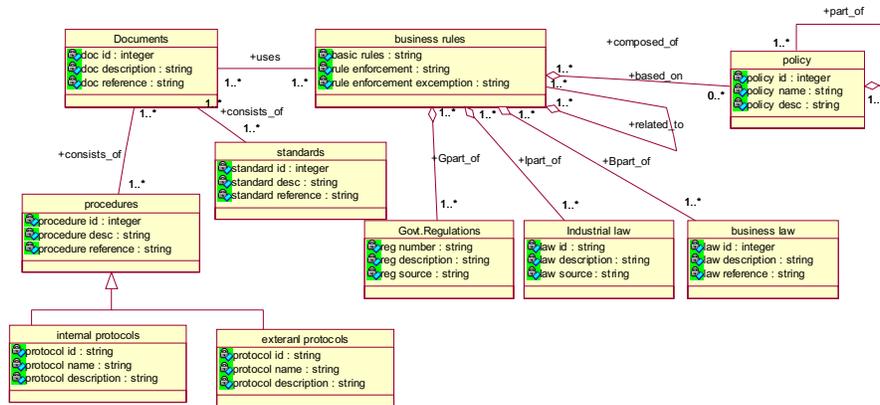


Figure 3: Business rules model

5. Benefits using our proposed model

The business rule driven platform technology drives the business processes. The system can automate many human decision making tasks by defining the business rules, instead of using traditional program development tools to code the logic of the task.

There are many benefits of using our proposed model in development of any business solution including:

- Easy to change business rules, without the need to re-think program logic or sequence of execution
- Rules are easy to understand for technical and non-technical users
- A Business rule helps computing the right action for the set of conditions and reducing complexity.
- IT and business analysts spend less time developing processes, thereby reducing overhead
- Changes do not impact business models
- Prevents business errors and mistakes
- Enables businesses to directly drive application functionality
- Enables a broad base of business analysts skilled in business problem solving to drive solutions

- Provide reduced total cost of ownership and eliminates bottlenecks to maintenance and improvement

This allows dynamic, incremental change to handle new needs.

6. An example for our proposed model

In this example we try to incorporate our proposed approach into existing business solutions. We are able to separate business rules and place them into component where this component makes relation with other components of a system. Through this example we analyse and able to customise business processes.

We consider an example of vehicle reservation system, where customers can reserve different kind of vehicles for particular data/time and return them. The reservation system involves many steps, but for the sack of simplicity here we only discuss its process and involvement of business rules into one simple business process. We discuss the scenario of returning damaged vehicle and then system will perform different checks and produced different outcomes.

Traditionally there are different components in vehicle reservation system, such as reservation, users, contract and invoicing component. The entire vehicle returning process will happen within these components. If we analyse carefully that in implementation all the business

rules, strategies come under ‘pre – post conditions’ or ‘if then else conditions’ format. For example “if customer will return damaged vehicle then what will happen”. The reservation system will bring up the signed contract, insurance arrangements, accident/ damage report, and police report.

For this particular situation a company will have their rules and policies, which may involve third party rules, such as involvement of police laws, and insurance company’s policies. The third party will verify these conditions based on their own rules and policies. The process of returning damaged vehicle is also depends upon other businesses.

Using our proposed model mention in section 4 where we separate all the business rules from implementation and place them into separate component called business rules component. This component may have sub components or packages. The sub components or packages may be called as internal or external rules. The internal rules are business own rules including policies, constraints etc, whereas external rules may include government regulations, industrial laws etc. In the following diagram we only represent component view of vehicle reservation system.

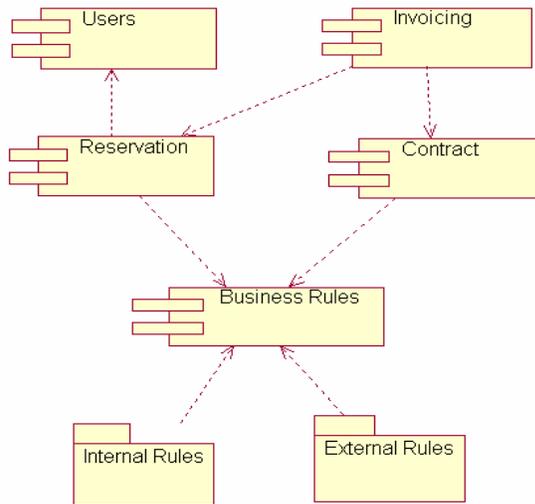


Figure 4: Components of Vehicle Reservation System

A business user is a generic term for customer, supplier, contact partner, and employee. A customer can only make a reservation. A customer may have any number of reservations, but a reservation always belongs to one particular customer. The contract is stipulated for the rental of a specific vehicle and for specific accessories. For pieces of equipment such as children’s seat, roof racks, and so on. Invoicing component consists of generating invoice, reports etc.

Each component has relationships between one another; they may or may not depend upon other components. For example rental contract may originate from a reservation. Since the contract does not assume a reservation, and the rental data may deviate from the original reservation data, several attributes of the reservation will also be contained in the rental contract. Deviations occur, for example, when a customer decides to return the vehicle earlier than originally reserved. Or a company gives the customer a better vehicle for the same price because the reserved type is currently not available.

During vehicle hand-over, a vehicle return protocol is drawn up which describes the returning state of the vehicle (existing damages, mileage, and so on). When the vehicle is returned, a corresponding return protocol is made. The difference, for example in mileage, is then taken as a basis for invoicing. If the vehicle was involved in an accident, an additional accident protocol must be drawn up, which is considered part of the return protocol.

Similarly to the reservation, the rental contract includes the agreed rental period. The returning protocols are then used to record the actual hand-over and return times.

The reservation system will check all the relevant business strategies, rules, and policies through business rules component. In this component business analyst will gather all the relevant information regarding business rules, and in future it will be easy to customise any business rules or incorporate other rules within this system.

7. System Architecture to Implement our Proposed Model

This implementation takes the steps to represents better and effective way to define business models using software engineering principles. The objective of this implementation is to illustrate the solution and proof of concept.

For the purpose of this implementation we use XML files, XSLT, Native XML database (Xindice), Java, Java Beans, JSP, XML Parser (Xerces), Xalan (XSLT Processor, for converting XML to HTML) and Tomcat application server. XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose.

The need to process and store XML has spawned several new types of software tool one of which is the “native XML database”.

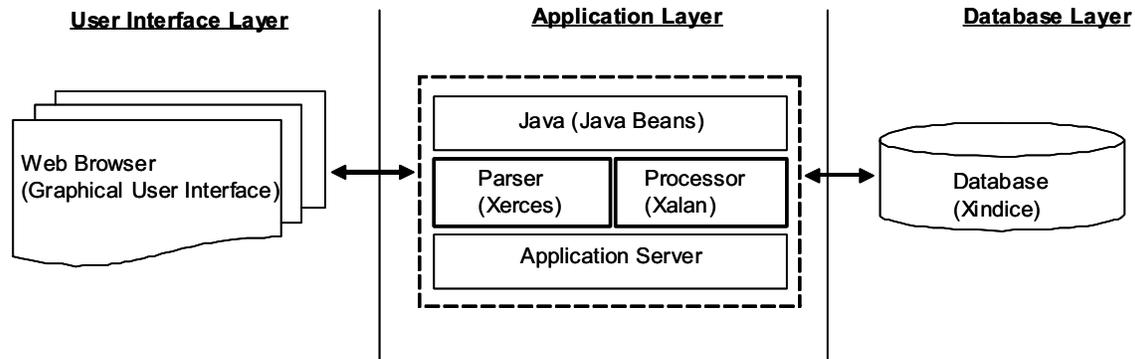


Figure 5: Proposed system architecture

Basically the system is composed of three tiers as shown in figure 5: an XML database (xindice from Apache) and Java on the backend, an application server in the middle tier, and a browser for the front end to display the results. The parser and processor's are part of the Java backend tier.

The user interface or presentation part is used for display purposes; user will interact with this part. The user selects necessary information, and the system will run the query to retrieve XML database from xindice. The Java Bean plays the role of a middle tier, which is responsible for database connectivity. The XML data then processed and transformed into XSLT format and represented in JSP pages on the web.

8. Application: a case study using our proposed model

We develop a very simple case study based on our model using current technologies. The model provides the flexibility by defining business rules by using software engineering principles.

We adopt the same case study as defined in our example in section 6. We only implement scenario of returning vehicle. The initial menu will allow users to perform some basic selections to access the system, due to space constraint we are unable to show basic screen shot in this paper.

The company can select different policies and their relevant procedures associated with a particular job. Any change or new implementation of these rules, policies and procedures, which are defined separately from the system logic, with change of rules, will not disrupt the whole business system.



Rentals & Rentals

All Types of Vehicle Rentals

255 Alexander Ave, 3069 Victoria, Australia
 18001900100, 00613987458, info@realrentals.com
 www.RealRentals.com

Customer Information:

Job Name	Renting Vehicle
Rental Date	2002 - 12 - 12
Customer Id	968574
Customer Name	John Smith

Job Description:

Job Status :	approved renting vehicle
Job Details:	approved 4wd renting vehicle
Rent To :	Westins
Rental office :	Real Rentals
Job Date :	2002 - 12 - 13
Remarks :	approved short term rental for customer John smith.

Company's Policies :

Policy Id	025703
Policy Name	Age Old
Policy Description	Customer must fulfil Procedures Pro 1, 2, 3 and Pro 4.

Company's Procedure :

Procedure	Procedure Name	Procedure Details
Pro 1	Must have current driving license	Show Details
Pro 2	Must required insurance cover	Show Details
Pro 3	Provide Reference of person or company	Show Details
Pro 4	Must required special permission	Show Details

Procedure Details:

Procedure id	Procedure Name	Procedure Description
02569876	Must have current driving license	Original or certified National and international licenses accepted
02569877	Must required insurance cover	Must Pay Vehicle Insurance for duration of hiring period.
02569878	Provide Reference of person or company	Reference of Person or company required
02569879	Must required special permission	Must provide special permission from guardian

In our implementation, we adopt component based development approach where business rules are implemented as a separate component. We defined business rules in the form of XML documents.

9. Conclusion & Future Work

Our modelling approach helps to visualise and analyse business processes at the most granular level, so that business people can see what needs to be done, by whom and when.

We produced end results by using our proposed model with use of software engineering principles. We also incorporate current technologies to implement our approach. The users will get the same end results as they used to it and also get great flexibility and reusable system.

For future work, we will introduce business process component and rules repository concept, where we use rules as a single common repository for storage and management of all business rules. These rules can be represented in XML-based format with benefits including collaboration between rules component and other business applications. This concept enables the use of rule repository to utilise business rules for modelling and analysis, while retaining their specialised environment. This central repository could be accessible by different applications or businesses.

Reference:

- [1] T.Torabi, S.Ali and W.Rahayu, "A Specification for Business Model Components for B2B Communication", in proceedings on IASTED Software Engineering Application Conference, Marina del Rey, CA, USA, 2003.
- [2] J.C.Leite and M.C.Leonardi, "Business Rules as Organizational Policies", in IEEE Ninth International workshop on Software Specification and Design, p. 68 - 76, Ise-Shima Japan, 1998.
- [3] T.Torabi, T.S.Dillon and W.Rahayu, "XML Schema for Software Process Framework", in 21st IASTED International Multi-conference on Applied Information, Innsbruck, Austria, 2003.
- [4] S.Aissi, P. Malu, and K. Srinivasan, "E-Business Process Modeling: The Next Big Step", in IEEE computer Magazine volume 35, number 5, p 55 -61, 2002
- [5] A.T.Silvia and F.Xavier, "Software Process Modelling", in proceedings on the software process in software engineering and knowledge engineering, Argentina, 2001.
- [6] Jamie Friedman, "The RulesPower Business Process Management Systems", in RulesPower White paper, pp. 1- 9, 2003.
- [7] Ellen Gottesdiener, "Capturing Business Rules", Software Development Magazine: Management Forum, Vol. 7, No. 12, December 1999.
- [8] Presley Becerra, "The living Transaction", in Intelligent Enterprise Magazine, Vol.4, No. 8, May 24, 2001.
- [9] Ivo Vondrak, "Business Process Modeling and Simulation for Quality Management", European Simulation Multiconference ESM 2000m SCS, Gent 2000, Belgium, ISBN 1-56555-204-0,pp.375-379. 2000.
- [10] N.R. Jennings, T.J.Norman, P.Faratin, P.O.Brien and B.Odgers, "Autonomous Agents for Business Process Management", Int. Journal of Applied Artificial Intelligence 14(2) pp.145 – 189, 2000.
- [11] A.Borgida, T. Murata, "Tolerating exceptions in workflows: a unified framework for data and processes", Proceedings of the international joint conference on Work activities coordination and collaboration. San Francisco, California, United States, ISBN: 1-58113-070-8, 1999, pp. 59-68.
- [12] A.Borgida, T. Murata, "A unified framework for exceptions in workflow and process models – an approach based on persistent objects", Proc. WACC, 1999.
- [13] J.Suzuki, Y.Yamamoto, "Extending UML with Aspects" Aspect support in the Design Phase", 3rd. Aspect-Oriented Programming (AOP) Workshop at ECOOP, 1999.
- [14] F.Duclos, J. Estublier and P.Morat, "Describing and Using Non Functional Aspects in Component Based Applications", Proceedings of the 1st. international conference on Aspect-Oriented Software Development, Enschede, The Netherlands, Apr. 2002. ACM Press, pp. 65 – 75.
- [15] B. Bryon, Business Modeling with UML: The light at the end of the Tunnel, Proc. Rational Edge, 2001.
- [16] E.R. Harold, S.W. Means, R.T. Erik, XML Hand Book, in: Amazon (Ed.), XML Hand Book, Vol. 1, 2001), 139 - 146.
- [17] M. Dubosson-Torbay, A. Lausanne, Y. Pigneur, eBusiness Model, Design, classification and Measurement, Proc. Thunderbird International Business Review, 2001.