

Integration of a Business Rules Engine to Manage Frequently Changing Workflow: A Case Study of Insurance Underwriting Workflow

George K Royce

Department of Information Systems and Quantitative Analysis,
University of Nebraska at Omaha
groyce@mail.unomaha.edu

Abstract

The purpose of this paper is to describe the challenges faced with the initial introduction and use of workflow for underwriting in an insurance company and how these challenges were overcome when the company was faced with the requirement to upgrade the system. Early workflow systems, including one of the original FileNet Workflo™ system, offered the promise of easy to change, work assignment and routing that could be managed by the business area. In the first implementation of workflow, the system did not provide the agility to easily change the system when the business requirements changed. When an opportunity was provided to update a legacy FileNet Image and Workflo™ system, the business management and IT decided to ensure that users were empowered to make their own changes in work assignment, reassignment and routing rules in the future by introducing a business rules engine to manage the workflow process. The case study details the approach used to implement a rule driven workflow system capable of “exception based underwriting” and provides lessons learned.

1. Introduction

Insurance companies have long been paper intensive companies using applications for insurance forms and claim forms just to name the most common. As a result, they have also been early adopters of image and workflow systems from companies such as FileNet and IBM as pointed out by Liberatore and Breen (1997). According to the Allen (2001), the Workflow Management Coalition defines workflow as “The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.” They further define a workflow management system as “A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.”

Prior to image and workflow systems used by insurance companies, staff used to hand out folders of applications for insurance to underwriters. These paper folders would be manually routed from clerk to underwriter assistant to underwriter. When the policy was finally issued, the folder and its contents (application for insurance, correspondence, medical tests, etc.) was sent to the files area of the company. The folder would also be used to hold additional correspondence between the company and the policy holder throughout the life of the policy. In the 1970's technology referred to as workflow systems was developed. One of the first workflow systems is described in a paper by Ellis and Nutt (1980). It describes the earliest workflow system developed at Xerox Parc by Clarence (Skip) Ellis. This original work gave rise to systems eventually marketed by companies such as Xerox, IBM and FileNet. These early systems could describe a business process in a workflow modeling language which could be changed by IT staff and later by skilled business analysts.

With the advent of image and workflow systems, personnel costs were reduced. Staff was no longer needed to route the paper files and manage the flow of work. However, early workflow systems have been criticized as inflexible by Grudin, 1988. Workflow systems may have begun with work queues and routing set up correctly but rapid changes in business soon required changes to routing rules and work groups. Early systems had a difficult time keeping up with the rapid pace of change in business. The early workflow systems used a business process modeling language. Often these were difficult to change by the business analyst and required the models to be reloaded so that changes could not be made dynamically. These difficulties are described by Aversano and Canfora (2002), and Agostino and DeMichelis (2000). The business was then required to invest in information systems staff to make frequent changes to the workflow system if they were not able to make the changes on their own. Some attempts have been made to create a more dynamic approach to changing work assignment and business process routing by more dynamically changing the business process models. This work has been described by Ellis, Maltzahn and Rozenberg, 1997 and by Jorgensen, 2001 among others.

This case study concerns an existing image and workflow system used by some 200 underwriter that was installed in 1995. The system was successful at eliminating paper routing. However, it required significant support by information systems personnel (IT) to make simple changes in the workflow system. An opportunity to replace components of the system due to a Visual Basic upgrade project led the business users and IT to consider how they could improve the next generation of workflow to ensure that it would be easy to use by business users and it would permit business analysts to make routine changes in routing rules and workflow groups so that they could more quickly react to business change at a lower cost with minimal IT involvement.

2. Background and Business Problem

2.1 Background

An insurance company in the Midwestern United States had implemented FileNet Images Services™ and the original FileNet Workflo™ in 1995 for the automation of image routing and workflow management in the Individual Life Insurance and Individual Health Insurance Underwriting areas of the company. The system provided for the scanning of applications for insurance and indexing of these applications in the system. Indexing is the process of adding key metadata to the image file so that basic information such as policy number, policy applicant, policy type, face value etc. is associated with the file. Once the policies are scanned and indexed, they are routed to a clerk to add detailed information into an insurance administrative system. The policy is then routed to an underwriting assistant to determine what information is needed to underwrite the policy. For simple policies, this may be just a check with the MIB (Medical Information Bureau) and perhaps obtain a Motor Vehicle Report. Once all the information has been collected, the underwriting assistant routes the file to an underwriter to determine if it should be issued and what riders might apply to the policy. This information is then entered into the administrative system and the policy is printed along with a copy of the original application for insurance and sent to the agent or broker to be delivered to the insured. If the policy is declined a letter is sent to the insured explaining the reasons for the decline. See Figure 1. The total number of underwriters, underwriter assistants and managers in the department has varied over the years but has been around 200 for most of that time. The technical architecture of the workflow system included a server based workflow system that communicated with a client running on Windows workstations written in Visual Basic 6. The image software was similar in that it consisted of the Images Services running on a UNIX server and communicating with an image viewer that was running on each client. This image viewer also contains components that were written in Visual Basic 6.

The initial project to eliminate the routing of paper documents through the underwriting department was considered a success and reduced the need for a number of full time equivalents that moved the paper through the organization and also required to track down policies that were not moving in a timely manner. This cost savings and additional reporting were the original benefits of the project. The project allowed the company to lower the cost of processing each policy and ensured that policies could be tracked. Time service on each policy was measured. However, one aspect of this original project did not work as planned. It was hoped that the business analysts would be able make routine changes in the FileNet Workflo™ as business requirements changed and new products were developed and underwritten. In general, the business analysts found this to be more complex than had been anticipated and the collaborative process between information technology staff and the business analysts had not been clearly determined. For these reasons, the information technology support team made essentially all changes to the workflow and image system throughout the years. In many cases, since the IT staff on the team was very familiar with Visual Basic and less familiar with FileNet Workflo™, they often added new functionality using Visual Basic. This meant that any time new groups were added or significant process changes were made, IT needed to make changes in Visual Basic 6.

2.2 Problems with the System

A project began in 2004 to begin reviewing all applications written in Visual Basic 6 since Microsoft had indicated that they were dropping support for Visual Basic 6 in favor of the managed development and run-time environment of Visual Basic .NET (Microsoft Life Cycle Support for Visual Basic). This also meant that the FileNet Workflo™ would also need to change to a new workflow system from FileNet or another company. The architecture that FileNet was now using was now J2EE and used a browser plug-in for image viewing and a browser interface for workflow. This essentially meant that much of the applications in place needed to be replaced with an upgraded image viewing facility and a browser based workflow system. In addition, the company was rolling out a producer portal for the companies thousands of agents and brokers. It was decided that the new interface would also be portal based so that the underwriters and underwriter assistants could eventually collaborate with the producers of the policies they were underwriting.

When IT approached underwriting management with the technology currency issues described above, they determined that the new system should address the following business issues.

1. Underwriting management was anxious to reduce the time it took to make changes to the system in the future. They also were accepting of the idea that perhaps routine changes that now must be done in IT should be done by trained business analysts.
2. There was a lack of good reporting on the status of a particular policy in the current workflow process. They needed to be more responsive to their agents and brokers and let them know where a policy was in the underwriting process at any time. There was a strong interest in providing that reporting on the producer portal for agents and brokers. The current reports that agents or brokers received about underwriting status were several days behind and were cryptic.
3. The underwriting management also wanted the new system to position the company for exception based underwriting in which a rule engine might “auto-issue” a policy without underwriter intervention if the decisions were relatively simple. The use of rule engines is common in property and casualty insurance and increasing in life and health insurance.

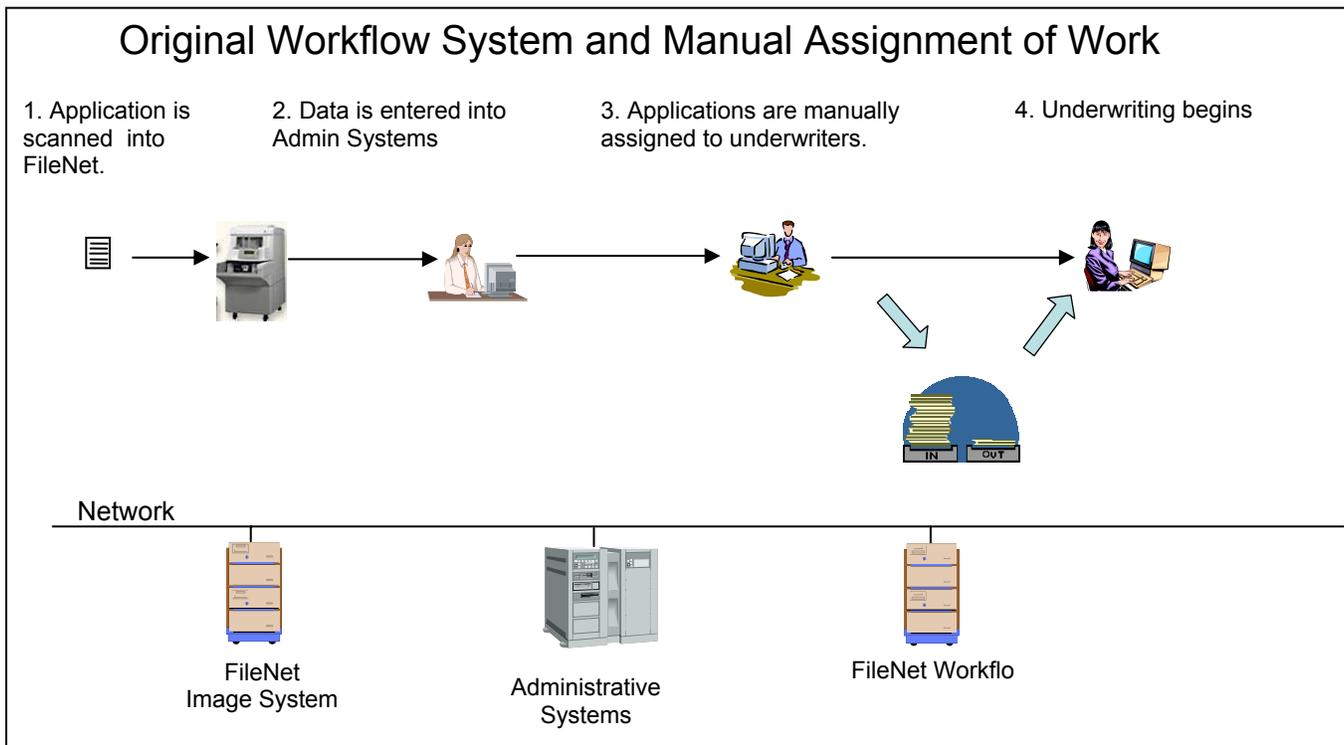


Figure 1

3. Requirements Overview

The new system needed to:

1. Replace the current dependency on Visual Basic 6 code and the original FileNet Workflo™ what was at the end of support.
2. Move to a browser based interface that was integrated with the IBM Websphere Portal Server for both image viewing and workflow so that collaboration between producers, underwriter assistants and underwriters can eventually be enhanced. The additional business value of the browser interface was that it enabled easier deployment to remote underwriters who may not be on the insurance company’s campus.
3. Empower the business analyst to make routine routing rule changes and workgroup changes without involving IT in the process. This would increase the business agility and reduce costs.
4. Empower the business to put up new products and underwriting groups in the system without IT involvement.
5. Integrate a business rule engine into the system in order to enable “auto-issue” of some forms of Individuals Life and Health insurance. This will help move toward the vision of exception based underwriting.

4. Pilots

With the significant size and scope of this effort it was determined that pilots might be useful to test the effectiveness of key components of the system before launching into a large conversion effort. Another reason for the pilots was to aid in the change management process. The introduction of a new workflow system with a rule engine and exception based underwriting is a significant change for an underwriting organization. It was hoped that these pilots would help the organization through the change process gradually. The first pilot was the use of the rule engine to make initial assignment of

policies to the underwriters. The current process of assignment was to ask the clerks indexing the policies to use a complex matrix of rules displayed on spreadsheets. The assignment rules involved a number of parameters such as “type of policy”, “region of country”, “face amount of policy”, “ability of underwriters” to determine the initial assignment of the policy for underwriting.

A sample routing rule is shown below:

If channel equals agency,
 and plan equals mortgage term or whole life,
 and region equals Midwest,
 and age is greater than 18 and less than or equal to 65,
 and face amount is more than \$250,000 and less than \$1,000,000,
 and policy is a conversion from existing policy

Then assign to Midwest Level 1 Underwriter Group.

These rules are often displayed in a rule table as shown below. These have been preferred by the business analysts

No	Channel	Region	Age	Face Amount	Conversion?	Assignment
1	Agency	Midwest	18<=65	250,000 <=<1,000,000	no	Midwest Level 1 Underwriter Group
2	Brokerage	East	18<=65	>=1,000,000	yes	East Level 2 Underwriter Group
3	Agency	South	18<=65	<250,000	no	Underwriter Assistant Group

Table 1

A small project was initiated to select a user friendly rule engine which could be used by the business analysts to do initial routing of the policies based on frequently changing rules. The business analysts were actively involved with IT in the selection of a tool which they could use to accomplish initial assignment. Criteria for selecting the rule engine included:

1. Rule engine must be easy to use for both IT developer and business analyst. Since most of the rule changes are to be done by the business analyst, the business analyst user interface and ease of use took priority over IT ease of use and user interface.
2. Rule engine must be able to integrate well with FileNet Image Services and integrate with IBM Process Server via web services or IBM MQ Series messaging.
3. Rule Engine must be certified to run in an IBM Websphere AIX (UNIX) environment.
4. Rule engine must have it own rule repository and ability to easily promote rules from systems integration environment to CAT (customer acceptance testing) environment to production environment.
5. Rule engine must have extensive tooling to diagnose and resolve problems as well as tooling to ensure good performance.
6. Rule engine needed to provide evidence of it's ability to scale to hundreds of transactions per minute based on future requirements.

The rule engines reviewed included Ilog, Blaze and Corticon. The Fair Isaac Blaze Rule Engine was selected since it best met the requirements outlined above.. It also offered a simple approach where by the IT staff could be there as a coach for the Business Analyst rather than do the work for them. IT and the business analyst could test the rules to understand the impact before putting them in place. This was installed in 2005 for the initial assignment of individual all policies underwritten by the company. The underwriting management was pleased with the approach since it allowed for quick changes in initial assignment with essentially no IT involvement.

The additional impact of this first pilot was that the underwriting area became accustomed to an easy to use rule engine. After some months, it was the business who suggested to IT that they use this same rule engine to auto issue one of the highest volume insurance products being underwritten. In late 2005 this was done. The rule engine was integrated with both the old FileNet Workflo™ and automated assignment system and the insurance administrative system through web service integration. This allowed the rule engine to act as a “junior underwriter” for these policies. The data submitted to the rule engine allowed it to issue the policy, decline the policy or route it to an underwriter for further review. During 2006 approximately 20% of this high volume insurance product was “auto-issued” using the rule engine. This set the stage for further integration into a rule driven workflow system described below.

5. System Design and Implementation

5.1 System Architecture and Design

Based on the customer requirements described above, the business analyst needed to easily make new workflow routes for future products the company would underwrite. The business analyst also needed the ability to develop new groups of underwriters and assistants for these new products. This included a database of metadata about the underwriters and a rule engine which provided assignment rules. The rule engine use would now be expanded to include reassignment rules. The system also needed to integrate with the IBM Websphere Portal server already in use with agents and brokers. It also needed to provide reporting on events in the system to both the underwriting management and to agents and brokers in the field. The system needed to provide both underwriters and producers status on items needed to issue a policy such as arrival of drug tests, APS (attending physician's statement), etc. The system had to be able to create reports that could securely be provided to the correct underwriting staff as well as the correct producers. Based on these requirements, the following technology was selected to meet the business requirements of this new workflow system:

1. FileNet Image Services Resource Adapter which provides browser access to images stored in a FileNet Image Services system.
2. IBM Process Server – IBM's latest workflow system which has integration with IBM portal and can be integrated with major rule engine vendors. It also allowed the creation workflows using a business modeling tool integrated with the development environment (Rational Application Developer).
3. Fair Isaac Blaze Advisor business rule engine for maintenance of routing rules and to provide auto issue capabilities for simpler decisions.

Additional technology used to implement the system that had been in place already included

1. FileNet Image Services
2. Websphere Messaging Queue and Message Broker used for communication with the mainframe based life and health administrative systems
3. Jacada User Interface Middleware (server based screen scraper). This software was also used to create web services out of key mainframe transactions.
4. IBM Websphere Portal for user interface and security management. (used in conjunction with LDAP directory).
5. Mercury Load Runner – this is a tool used to load test systems to ensure that they can meet the performance standards required by customers.

Services of the life and health insurance administrative systems would continue to be used but their role in the issue process will be reduced. The new workflow system combined with the rule engine will become the hub of the system and the administrative systems will receive instructions to either print the policy and begin the billing processes for the policy or decline the policy and print the appropriate letter explaining why. This is consistent with the service oriented architecture of the new system in which the administrative system is no longer the hub of the system but simply provides services to the new underwriting workflow system. See Figures 2 and 3.

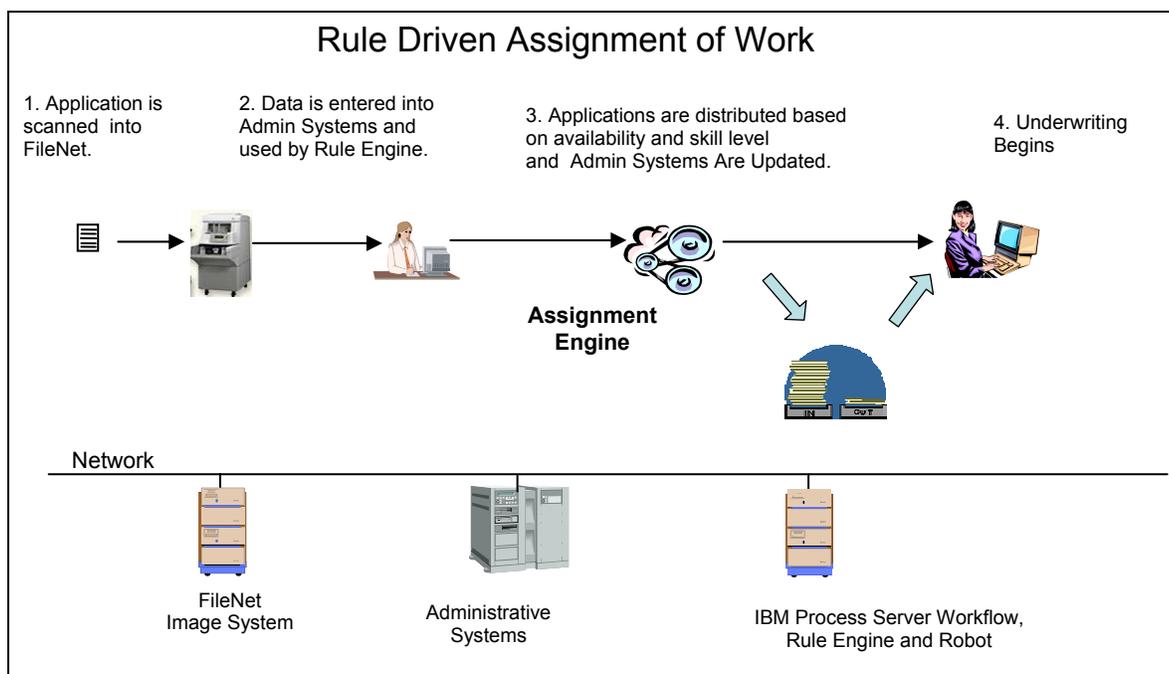


Figure 2

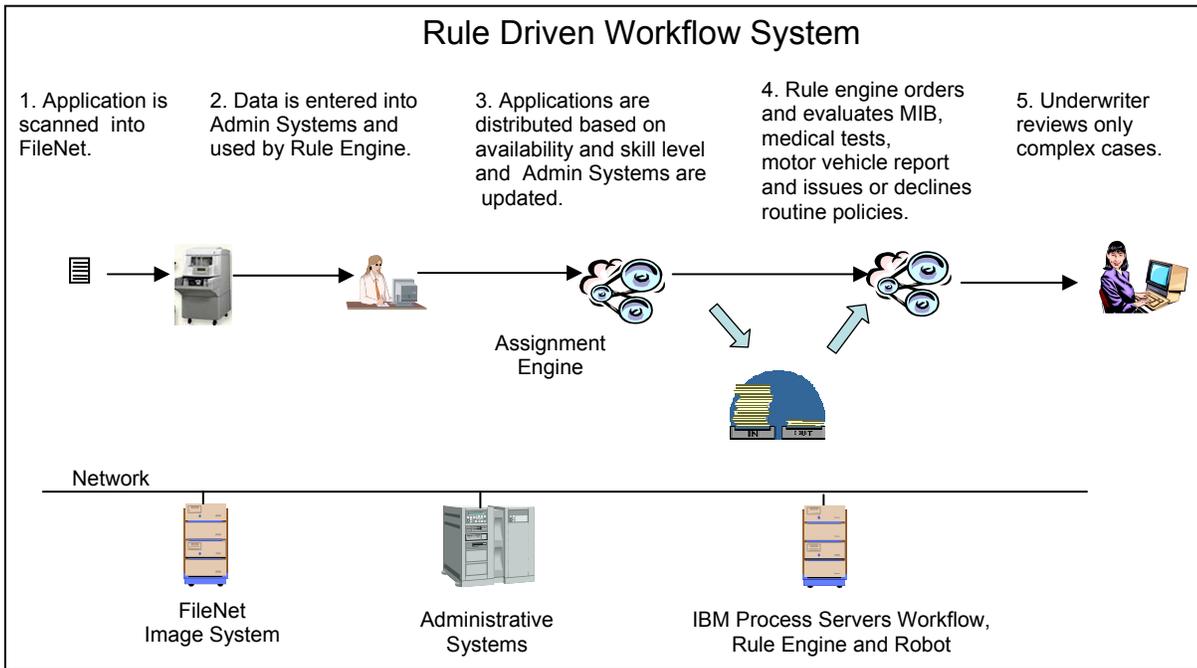


Figure 3

Figure 4 outlines the technical design of the rule driven workflow system. The most of the interfaces between the components use web services or IBM MQ Series Messaging.

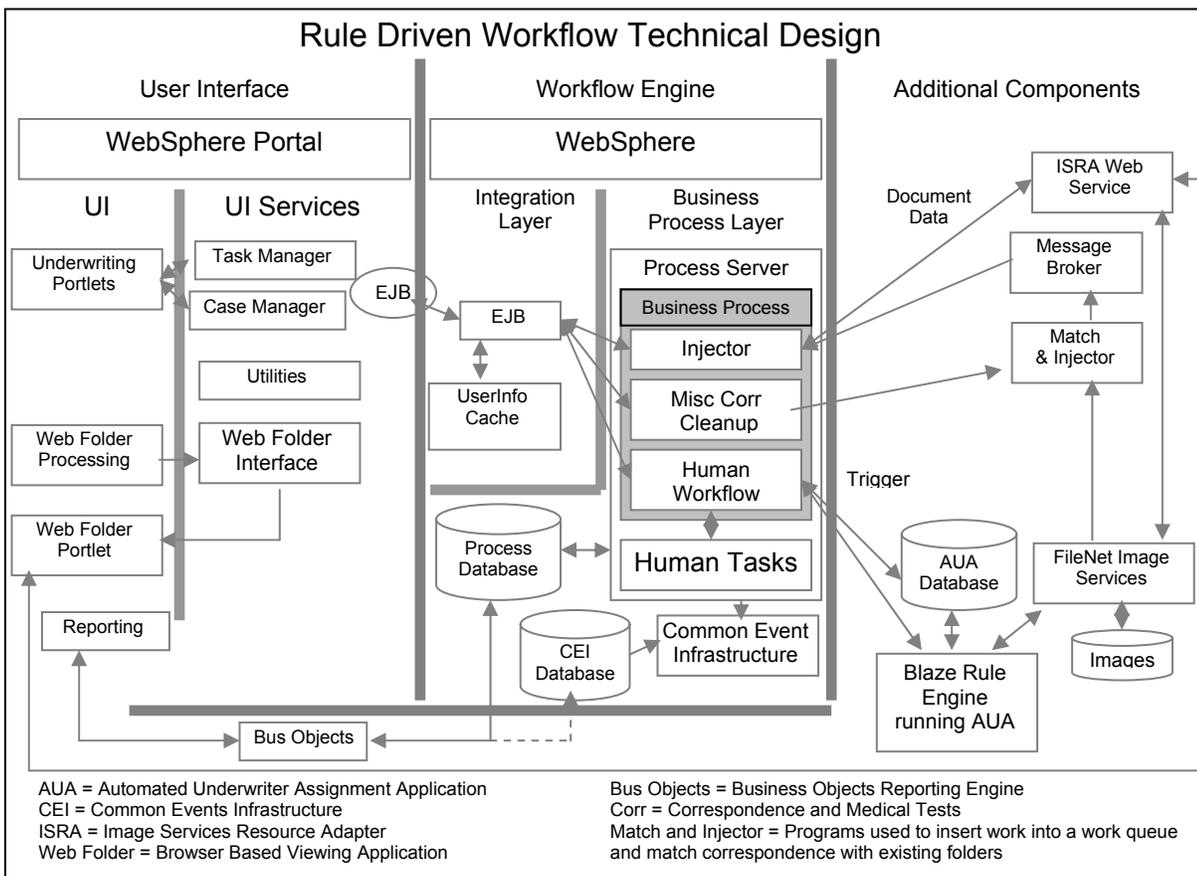


Figure 4

5.2 Migration approach

One of the dilemmas faced by the project team was the classic upgrade question “How do you change a tire on a car while driving at 60 miles an hour?” All work was being indexed and sent to a queue and that work was then distributed to

underwriters or assistants based on the recently added Underwriter Assignment application described above. The project team needed to allow work to be routed to both the old workflow system and the new workflow system until all work was accounted for in the new workflow system. The team would then shut down the old workflow system. This allowed for a gradual cut over to the new system which was the preferred approach of both the project team and of underwriter management. To accomplish this, after work was indexed, it was sent to the new workflow system via a web service. If it was the type of work not handled by the new workflow system, it was immediately rerouted to the old workflow system.

The project team started with low volume, simple life policies to ensure the system was functioning as planned. It also positioned the organization to eventually take advantage of auto-issue underwriting of these and other simple life policies using automated information gathering from online, secured web service available information such as the MIB (medical information bureau reports) and motor vehicle reports that are available to underwriters if the insurance applicant signs a form allowing the use of this information. This also allowed the project team and business analysts to work out the routing rules on simple policies first before tackling more complex insurance products.

One of the challenges faced by the team early in the testing of the new system is that the portal/browser based workflow and imaging viewing application both provided more information for the underwriters and others using the system. However, this meant that more of windows used by the underwriter overlapped and this often meant scrolling to see information. This would have lowered the productivity of the underwriters and others using the system. Based on these early tests, the project team recommended an upgrade the current 20 inch monitors to a wide screen flat panel display with a width of 24 inches and height of 16 inches for some underwriters. They had considered two montors but the cost of two monitors and video adapters was similar to one larger monitor and underwriters found it easier to use. Additional usability testing was conducted to ensure that both the browser based image viewer and work list portlets were easy to use by the underwriters and that they could navigate between portlets at least as efficiently as they did with the previous Windows user interface. In addition to the functional and usability testing, extensive load testing was conducted to ensure that the system could handle the growing volumes of life and health insurance the company was anticipating over the next 24 months. The Mercury Load Runner tool was used to simulate several hundred users accessing the system for various transactions at the same time to ensure it could handle the load and provide adequate response time.

The new underwriting system began handling its first production life insurance policies in January of 2007. Approximately 10 underwriters and underwriter assistants were involved in this initial cut over. There were minor issues with the routing of work between the two workflow systems in the first week which were quickly resolved. The customer response after one month of operation is positive. The business analysts are able to make changes in the business rules as necessary. The productivity of the underwriting staff has been positively impacted by the change for this initial form of policies. The project team is now working with the business analysts to add the routing and processing rules for the remaining life policies. Following that effort, the health policies will be added to the system and the old workflow system will be shut down and decommissioned.

6. Discussion, Lessons Learned and Suggestions for Research

6.1 Discussion

The project will be ongoing throughout 2007 but is already being viewed as a success in the eyes of underwriting management. The underwriting management sponsor has stated that the project has positioned us for the future by auto underwriting more transactional cases leaving more complex cases to underwriters. The new system is delivering on the goals that had been set out for the project on time and within budget. The system:

1. Replaced the client server Visual Basic 6 code and older FileNet Workflo™ with a browser based portal application that can more easily deployed to both home office and remote underwriters.
2. The system is more agile in that business users can make changes to routine routing information in an easy to use rule engine. It is also a rule engine that can be used to increasingly “auto-issue” simple policies with minimal help from IT. The IT staff have become coaches for the business analysts rather than coders for routine changes. Based on this success, more “auto-issue” work is now planned for the new system. This allows underwriting management to realize their vision of except based underwriting.
3. Reporting of underwriting business events is now much easier using this system. The reports are being used by underwriter management today. Once they become comfortable with the accuracy of the reports, this information will be shared through the portal to producers (agents and brokers).
4. The service oriented system now positions the company to be more responsive to selling and underwriting through multiple sales channels.

As portal based business process management systems with integrated, rule driven workflow begin to drive the work of areas like underwriting, claims and customer service the role of legacy systems in insurance companies will change. These “stove pipe” business applications such as life and health administrative systems, claims systems and customer service

systems will continue to provide value for the key unique services they do well such as billing, claims adjudication, etc... However, increasingly any application specific workflow in these systems will be either turned off or disregarded as the workflow is driven more by business process management systems that are managed by the business users in the future. This also positions the company to eventually replace these systems with the next generation service based systems.

One question that needs to be addressed here is why use a rule engine for routing rules rather than place these detailed rules in the process modeling tool that is provided by IBM Process Server and other workflow systems. There are two primary reasons for this.

- First, the business analysts have become very comfortable with the user interface of the rule engine they must work with for initial assignment and now the more complex auto-issue rules. They have a test environment which allows them to test these rules before moving them into production. This ease of use factor is a major reason that the rules were designed to be in a rule engine rather than in a complex business process model.
- Second, there is a lack of “round-trip modeling” capabilities today in even the latest generation workflow systems. This means that once a complex workflow model is developed using a workflow modeling tool like Websphere Business Process Modeler, the Business Process Execution Language (BPEL) and it’s with proprietary extensions is imported into the development tool used by the application developers. The IT staff complete the integration of the system and then deploy the workflow system to the production server. When changes are made to this model, the server needs to be recycled (usually at off-hours when no work is moving through the system). The problem with this approach is that it requires more involvement of IT staff and frequent recycles of the workflow server. By placing the routing rules in the rule engine, both of these issues are avoided.

The major workflow vendors appear to understand this need for more flexibility in their workflow systems and are embedding dynamic rules engines in their workflow systems and/or providing for integration with major rules engines such as Ilog, Corticon and Fair Isaac Blaze Advisor.

6.2 Lessons Learned

One of the primary lessons learned deals with the original workflow system. When it became clear that the business analysts were unwilling to make changes in the FileNet Workflo™ System, more training should have been conducted with the IT staff to ensure that they always had staff capable of supporting and making changes to the workflow system rather than tools they already knew (in this case Visual Basic). While this may not have provided the agility that is currently being delivered by a rule engine driven workflow, it would likely have been easier to make some routine changes.

The project has been successful at delivering an agile system which can be quickly changed. However, when two tools are used to solve a business problem, it is often difficult to determine which one to use. When should rules go in the rule engine and when some routing rules can stay in the business process model? As a best practice, the project team determined that if the rule is likely to change in the future (based on past experience), the rules should go in the rule engine or in data tables accessible by the business user. If the rules rarely change, it is more likely to stay in the process model only.

Keys to business analyst buy-in to manage their own changes were also learned by the project team:

- The user-interface of a rule engine and business process modeling tool must be intuitive
- The business analyst needs an IT coach with a both teaching and communication skills as well as a strong proficiency in rule engines and workflow.

6.3 Suggestions for Research

As mentioned in the lessons learned, more work is needed to determining when to use a rule engine and when to leave the rules in a business process model. Since many of the major workflow vendors are now integrating rule engines into their workflow system, this will become more important as IT and business analysts struggle to determine where to place rules.

Opportunities also exist in studying the collaboration between IT business rules coaches and business analysts. There is likely to be a growing need for more IT coaches as more systems are built with rule engines integrated within the system.

References

Agostino, A and DeMichelis, G “A Light Workflow Management System Using Simple Process Models,” Computer Supported Cooperative Work, vol. 9, no. 3-4, 2000.

Allen, Rob “Workflow: An Introduction,” Workflow EfMC External Relations Committee, 2001.

Aversano, L. and Canfora, G. “Introducing eServices in Business Process Models,” SEKE 2002, July 15-19, 2002, Ischia, Italy.

Ellis, C. A. and G.J Nutt “Office Information Systems and Computer Science,” ACM Computer Surveys, Vol 12, No 1 (March, 1980).

Ellis, C. A. and Maltzahn “The Chautauqua Workflow System,” Proceedings of the Thirtieth Annual Hawaii International Conference on Systems Sciences ISBN 0-8186-7862-3/97 March 1997.

Grudin, Jonathon “Why CSCW Applications Fail: Problems In The Design And Evaluation of Organizational Interfaces,” Proceedings of the ACM CSCW88 Conference. August 1988.

Jorgensen, Havard “Interaction as a Framework for Flexible Workflow Modeling” GROUP 2001, September 30-October 3, 2001, Boulder Colorado, USA.

Liberatore Mathew J. and Breen, Donna “Adoption and Implementation of Digital-Imaging Technology in the Banking and Insurance Industries,” IEEE Transactions on Engineering Management, (44:4), November 1997, pp 367-377.

Microsoft Life Cycle Support for Visual Basic, <http://support.microsoft.com/lifecycle/?p1=2971>