AN INTEGRATED OPEN SOURCE MINE WORKERS COMPENSATION SYSTEM

Louis Coetzee, Ronell Alberts, Hina Patel, Laurens Cloete[*]

ABSTRACT

This article describes the Mine Workers Compensation System developed by the CSIR and Molepe Consulting for the South African Department of Health. Mining activities increase the risk of certain occupational lung diseases. South African legislation governs the monitoring, surveillance and evaluation of miners for possible compensation due to occupational lung diseases. This article describes the various problems currently affecting the compensation process for current and former mineworkers suffering from these diseases (the compensation process is driven by independent and loose standing units interacting via article-based folders resulting in poor service delivery). The article describes the iterative and incremental methodology used in developing an integrated solution and contrasts this methodology with the typical waterfall approach (which very seldom works) widely used in other large software development projects. The integrated solution developed by the CSIR based on open-source software and concepts is presented. This solution is based on an innovative state based task engine and role-based access control. Finally the article presents the experiences gained, the challenges experienced and lessons learned through using open-source software in solving this challenging problem.

INTRODUCTION

South Africa’s economy has long been dependent on the mining industry. Unfortunately, mining activities increase the risk to the mineworker of contracting certain occupational lung diseases. Typical diseases include tuberculosis, obstructive airway disease, asbestosis and progressive systematic sclerosis. The health risks associated with working in the mining industry were identified at a very early time. Government introduced legislation (the Occupational Diseases in Mines and Works Act (ODMWA) 1973) that governs the lifelong monitoring, surveillance and evaluation of both former and active miners for possible compensatable occupational lung disease.

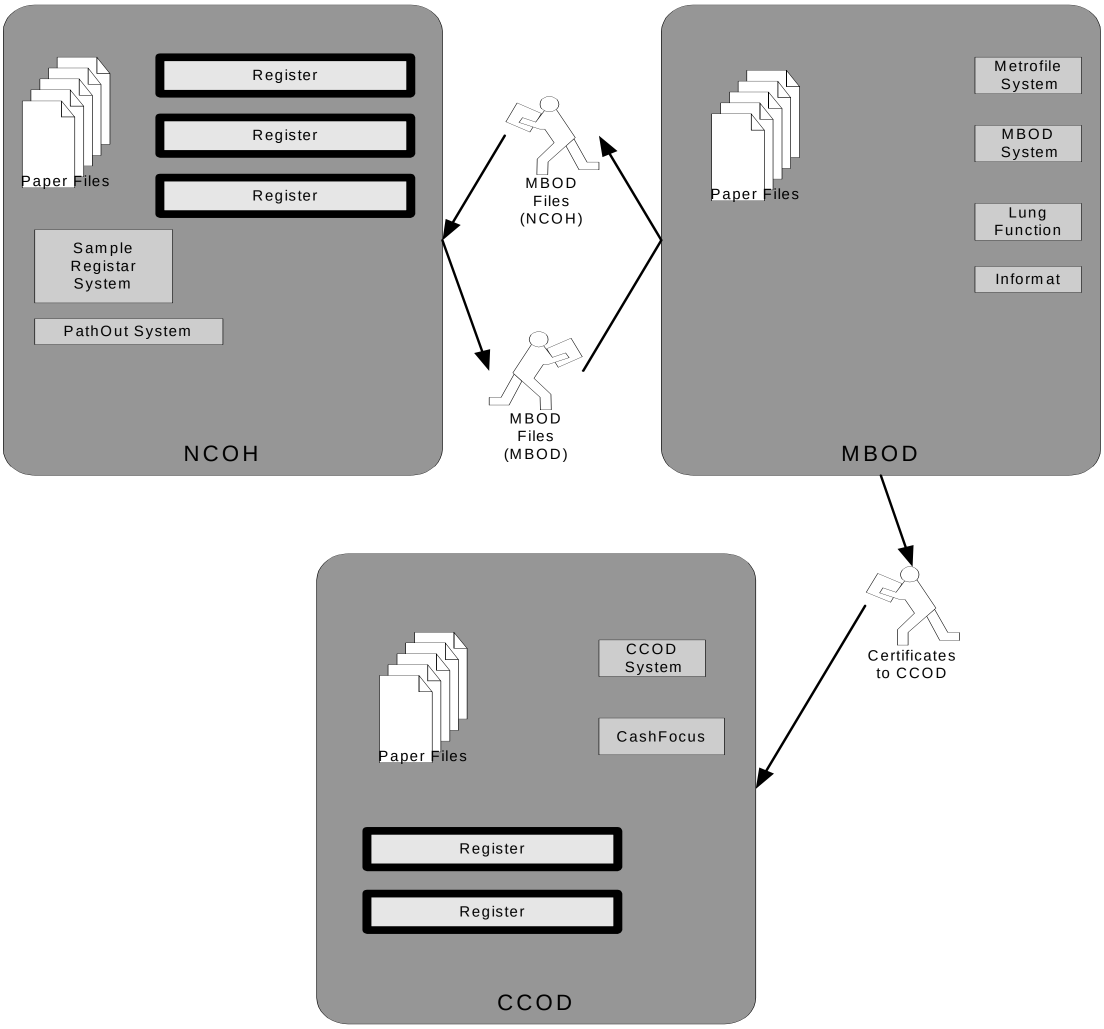
[*] Louis Coetzee, Ronell Alberts, Hina Patel, and Laurens Cloete are researchers at the CSRI.

This article describes a project aimed at improving service delivery to mineworkers and their relatives through increased effectiveness of internal government systems, followed by a description of the current reality, and the implemented solution (New Reality). There is also a description of the methodology used, the architecture implemented as well as the final solution. The next section relates the experience gained from implementing such a solution using open-source technologies, and finally, conclusions about the project.

Current Reality

Three different units in the South African Department of Health manage the mineworkers' compensation process: Medical Bureau for Occupational Diseases (MBOD), National Centre for Occupational Health (NCOH), and the Compensation Commissioner for Occupational Diseases (CCOD). MBOD drives the benefit medical examination and certification of a worker, while NCOH provides the statutory autopsy service required by the Act. CCOD is primarily concerned with the compensation process. FIGURE 1 depicts the current structure, support systems and associated process flow.

FIGURE 1 CURRENT REALITY (NON INTEGRATED)

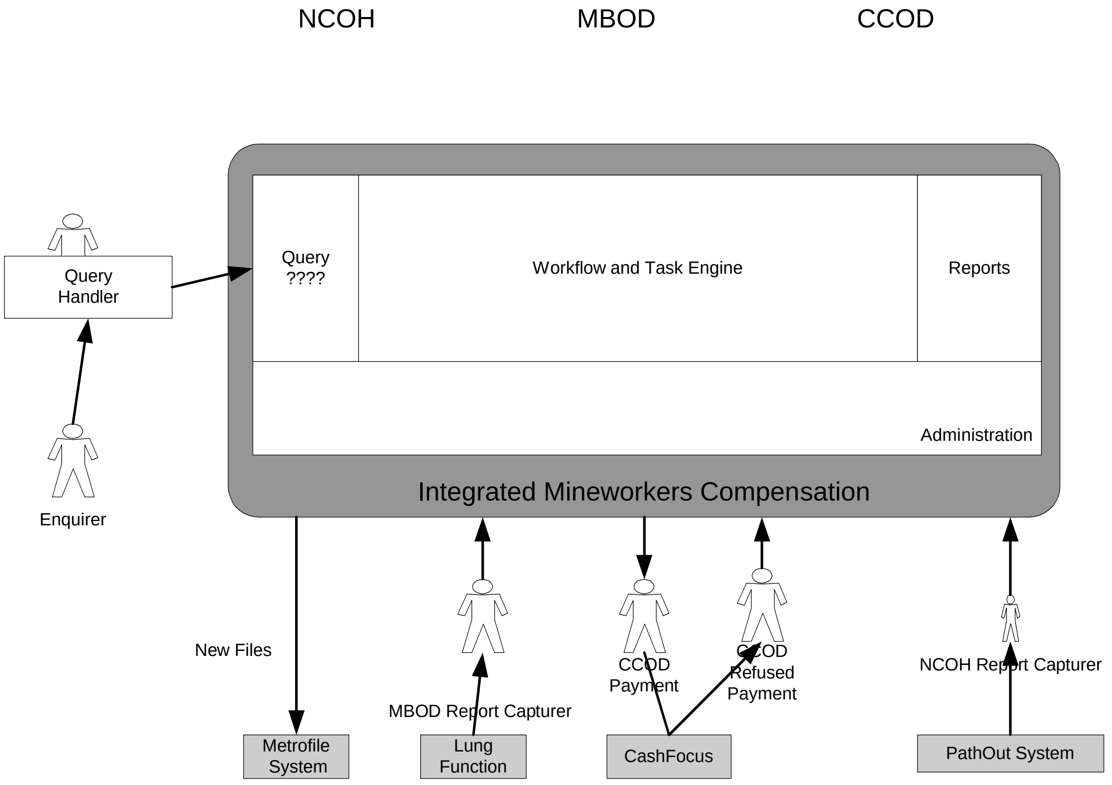


Each unit has loose standing support systems and procedures. In this reality each unit operates in a completely independent fashion. This results in duplication of effort as well as information (e.g. gathering of personal and contact information). In addition, each unit has to maintain costly infrastructure (e.g. database and application servers) as well as separate applications in order to manage the process. This process is inherently distributed whilst being driven by manual paper based activities. This leads to difficulties in claim status tracking as well as severe bottlenecks, resulting in poor service delivery in the form of delayed (or no) compensation to eligible applicants.

New Reality

In order to significantly improve service delivery the envisioned solution was based on an integrated compensation process that eliminates information duplication, reduces the number of manual forms and other unnecessary administrative activities, enhances security and minimizes the threat of fraud. In addition, the solution provides management with reports to evaluate the effectiveness of relevant business units, speeds up the compensation process time and provides the ability to respond to enquiries from applicants. FIGURE 2 illustrates the new integrated reality where the three business units use a centralized infrastructure that provides interfaces for business unit specific services. The *workflow and task engine* integrates all relevant business processes present in the three units. The *query* facility enables better tracking and management of status queries linked to claims. The *report* facility provides an interface to monitor and evaluate the efficiency of the services delivered. The *administration* facility provides an interface to efficiently manage users and their associated roles and responsibilities. In addition the system incorporates outputs from existing systems (e.g. CashFocus, Metrofile, Lung Function and PathOut) used in the environment to enhance efficiency.

FIGURE 2 NEW REALITY (INTEGRATED)

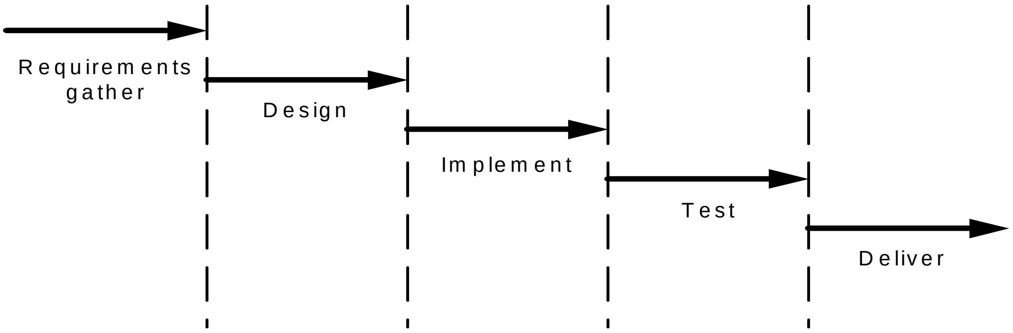


METHODOLOGY

Most often, large software systems are implemented using the linear waterfall methodology (

FIGURE 3) (Boehm, 1987 p.513): Requirements are captured that are signed off by the client before development commences. The system is developed based on the captured requirements. After all development has been done the system is tested and then delivered. Only at this point does the client have access to the developed system. This has been the preferred approach in many projects as it allows for easy planning and management and provides a sense of security to the developers through the sign-off of requirements. Unfortunately the desired outcomes are often not realized as the waterfall approach lacks flexibility and limits innovation. The waterfall lifecycle makes it very difficult (if not impossible) to implement the true requirements.

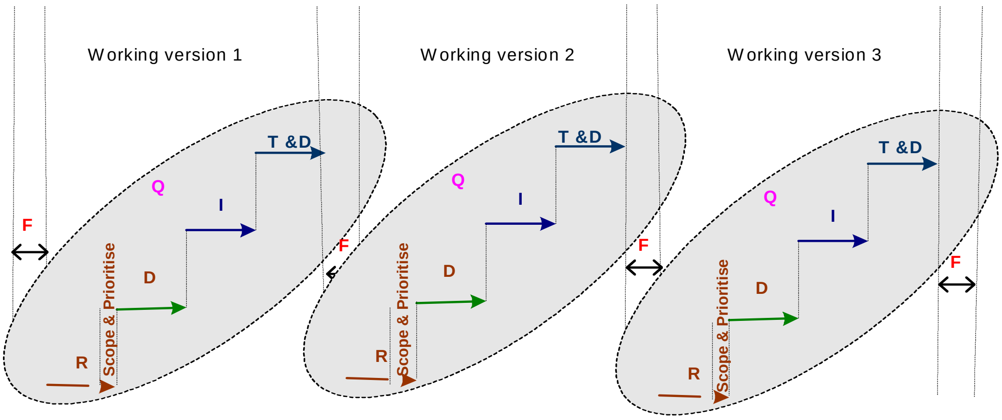
FIGURE 3: WATERFALL DEVELOPMENT METHODOLOGY



To ensure that the system really meets the requirements of all three departments an iterative and incremental methodology was followed (Larman, and Basili 2003, p.47). The development is grouped into smaller, more manageable chunks of work (the iterations), with each chunk containing a requirement analysis, design, implementation as well as testing and deployment phase. After every iteration the clients have access to the solution and interact with the system. This allows the clients to better understand their own requirements, which are fed back to the analysts and system architects. This is done in an interactive way, as it is very important to maintain close communication throughout the project lifecycle for this method to be successful. Using these new requirements, a next implementation phase commences. This iterative process is repeated until the required functionality is implemented in the desired way.

FIGURE 4 depicts this approach to developing a project.

FIGURE 4: ITERATIVE AND INCREMENTAL DEVELOPMENT METHODOLOGY



R - Requiremensts Analysis

D - Design

I - Implementation

T & D- Testing & Demo/Deploy

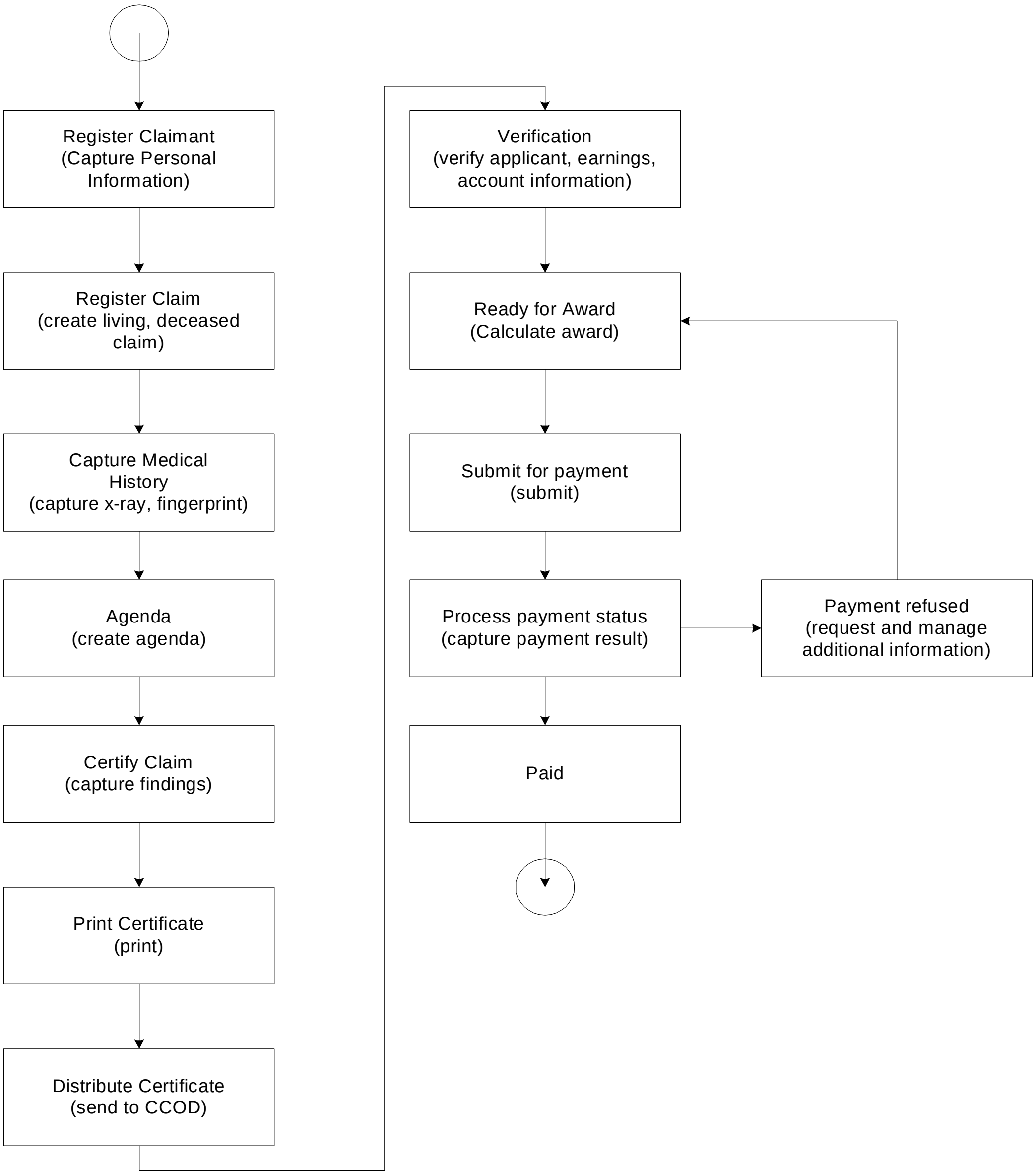
F - Client feedback & incorporate into system

Q - Quality Assurance

APPLICATION MODELLING

The integrated business process was modeled using a state diagram abstracted in the persistence layer and implemented as a task-engine in the application. Each state has various tasks that need to be completed before an application can progress to a following state. FIGURE 5 depicts a sample process flow for a claim. The various tasks applicable to each state appears on the user's task list (filtered by the user's assigned roles) allowing for fast and accurate claim progression. Abstracting the business rules in this way allowed for faster development in conjunction with the iterative and incremental methodology, as each iteration only required changes to the defined process in the database and not to the task-engine itself.

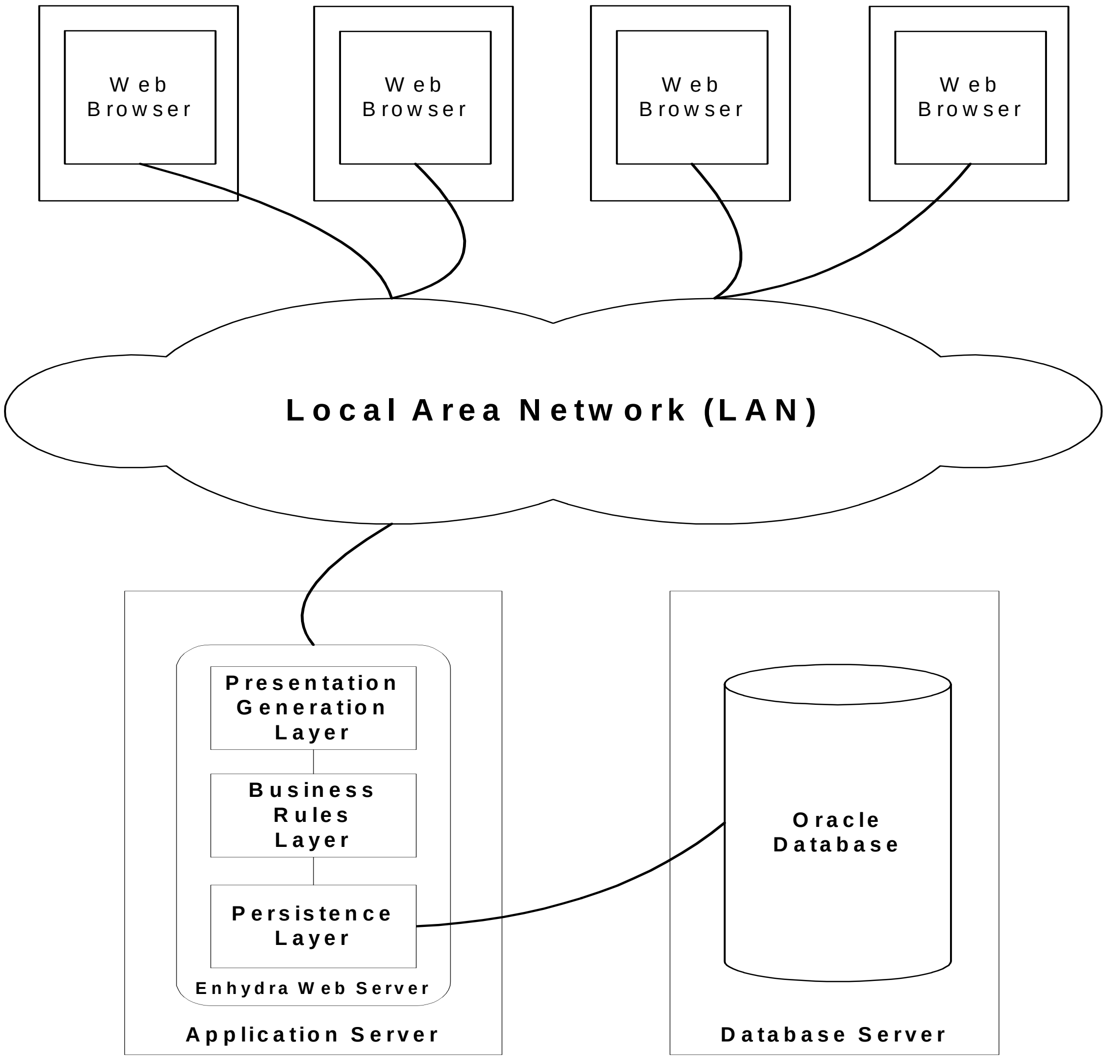
FIGURE 5: CLAIM PROCESS



ARCHITECTURE

The application was implemented using a three-tier architecture consisting of a presentation layer, business layer and a persistence layer. Separating an application in this way allows for easier maintenance and future enhancements. In order to avoid costly client installations an HTML-based presentation layer that is rendered by a client side web browser was chosen. The application generates the presentation layer based on business rules. The application accesses the persistence store to retrieve and update required information. The system architecture is shown in FIGURE 6.

FIGURE 6: SYSTEM ARCHITECTURE



IMPLEMENTATION

A large number of commercial possibilities exist that can be used to implement the chosen architecture. These include proprietary solutions based on Microsoft's .Net initiative as well as various commercial J2EE Java based application server solutions. The initial costs as well as yearly maintenance and support of these products tend to be prohibitive. Very often these commercial products are not open-standards based which increases the complexity when integrating with legacy systems. The unavailability of source code as well as dependence on the vendor for (often irregular) updates further complicates the matter.

Based on the above-mentioned constraints a decision was made to implement the solution using open-source technologies. Previously the department had already invested heavily in Oracle as persistence store. It was therefore decided to continue with Oracle as back-end store, even though open-source databases (PostgreSQL or mySql) would have been sufficient.

Various open-source technologies exist that could be used to implement a solution like this. These include Apache (Apache Software Foundation) combined with PHP as well as Java based application servers such as Tomcat (Apache Jakarta Project) using Java server pages. These technologies do not enforce a clear separation between logical layers that ultimately impacts the application's maintainability and extensibility.

The application was implemented using the open-source Enhydra application server (Enhydra). Enhydra is layered on top of Tomcat, thus providing additional functionality including improved session management. Enhydra also provides various tools that allow for the clean separation between the presentation layer and the business layer through its XMLC tool set. The framework includes an object-oriented application layer that greatly reduces development and maintenance effort. Connection to the persistence store (the Oracle database) was obtained using JDBC.

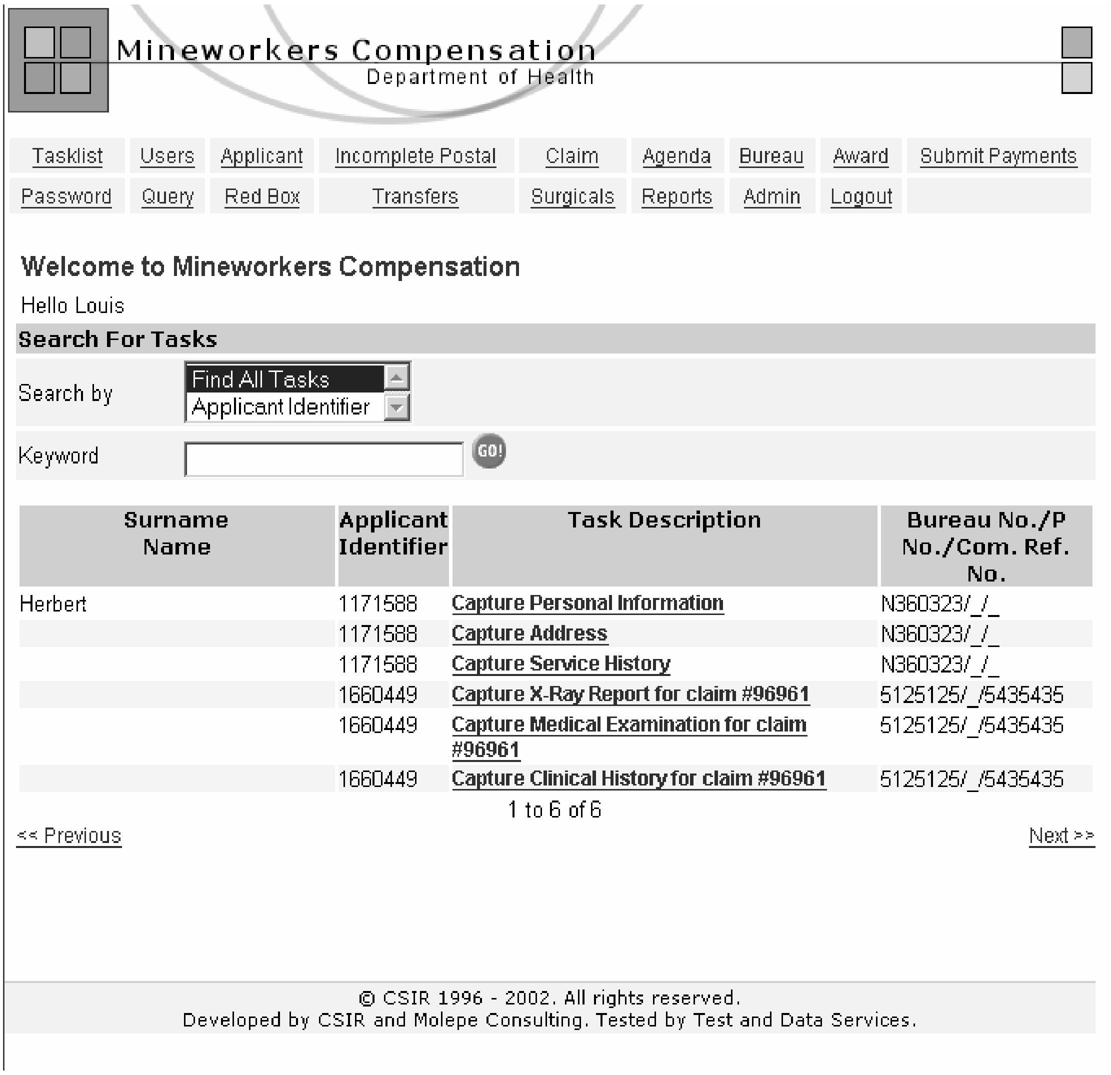
To support collaboration between the distributed users (each with different tasks and requirements) a role-based access control module was developed. Each registered user has various roles assigned, which in turn determines the various menu options available to that user. A task-engine was implemented that displayed only applicable tasks (based on the roles assigned to that user as well as claims in a specific state) to the user. The task-engine deskills the compensation process by leading the user. It enforces strict adherence to the business rules, thus preventing exploitation of loopholes and thus reducing the opportunity for fraud. Because the system is automated and seamlessly integrates over the three business units, the processing time of claims has improved resulting in faster compensation.

In addition, the integrated solution allowed for the implementation of a query module to provide applicants with the ability to query the status of their application. An operator logs a query, which stays open until resolved by the operator, or the person to which the operator has transferred the query.

FIGURE 7 depicts a task list for a user with many roles. By clicking on a specific link for a task, the user is taken directly to the appropriate form. Once this task (the form) has been completed additional tasks may appear on the user's task list page, depending on whether all tasks in the current claim state have been completed.

The figure also shows other links (Users, Applicant, Incomplete Postal, Claim etc.) Using these links the user can navigate to a specific form for a specific applicant if additional information needs to be entered. The visibilities of these links vary according to the assigned roles, thus preventing user overload (information overload without the desired background knowledge) as well as fraud, as a user cannot navigate to a page without the appropriate role.

FIGURE 7: USER TASK LIST



Welcome to Mineworkers Compensation

Hello Louis

Search For Tasks

Search by: Find All Tasks / Applicant Identifier

Keyword:

Surname Name	Applicant Identifier	Task Description	Bureau No./P No./Com. Ref. No.
Herbert	1171588	Capture Personal Information	N360323/_/_
	1171588	Capture Address	N360323/_/_
	1171588	Capture Service History	N360323/_/_
	1660449	Capture X-Ray Report for claim #96961	5125125/_/5435435
	1660449	Capture Medical Examination for claim #96961	5125125/_/5435435
	1660449	Capture Clinical History for claim #96961	5125125/_/5435435

1 to 6 of 6

<< Previous   Next >>

© CSIR 1996 - 2002. All rights reserved.
Developed by CSIR and Molepe Consulting. Tested by Test and Data Services.

Open-source Technology Experiences

Open-source technology is normally associated with very skilled IT resources and a hacker culture. As such the use of open-source technologies is not without risks and challenges. These are mostly centered on the continued support and maintenance of any application that is developed using these technologies. The Mine Workers Compensation system has a very long life expectancy. This implies that the technologies used also need to be maintained and supported for the duration of the application's life span. However, there is no guarantee that such a building block will have the supporting infrastructure for this to be the case. This risk can be negated through careful choice of the technology using criteria such as compliance to standards, the size and activity of the community involved with the building block development and track record of this community.

A major challenge in using any open-source technology is in maintaining an in-house knowledge base of that technology in order to provide normal application maintenance. The developers involved in the initial development of the application very seldom stay locked to that technology (part of the hacker culture), but often prefer to move on to the newest and most popular open-source building blocks, thus leaving a void in terms of maintaining knowledge of that particular technology.

The department's main concern is service delivery and in this case specifically compensating mineworkers. The team therefore had to reduce risk of using an open-source based approach by implementing a user friendly and low maintenance IT system. This was done by developing the role-based task-engine pushing various tasks to users determined by the current state of a claim and the individual roles assigned to the user. Extensive training, good communication as well as a substantial familiarization time also eased the transition to the new system and improved the acceptance rate. These factors combined with the successful deployment provided a very positive experience. The reduced upfront software purchase cost and avoidance of paying expensive license fees allowed the CSIR to develop the application for the Department of Health at a fraction of the cost of a proprietary system. The maturity, robustness and stability of the open-source technology used also helped the CSIR to provide an overall stable solution that further enriches the end-user experience of enhanced service delivery.

CONCLUSION

This article presented the integrated Mine Workers Compensation system developed to enhance and optimise the compensation service provided by the Department of Health. The project achievements can be summarized as follows:

- Successful implementation of an integrated Mine Workers Compensation system using open-source building blocks.
- The capture of the tacit knowledge of the Department of Health employees in a system that implements the full compensation process.
- An iterative and incremental approach was successfully used to implement a government system.

In future the system could be ‘generalized’ in order to be applicable to different problems with a workflow component. In conclusion: The authors believe that the open-source option should be considered when implementing government systems and that the iterative and incremental development methodology should be encouraged.

BIBLIOGRAPHY

Apache Software Foundation. (http://www.apache.org/)

Boehm, BW. 1987. A spiral model of software development and enhancement. *System and Software Requirements Engineering.* 513-527. IEEE Computer Society Press Tutorial.

Enhydra. (http://enhydra.enhydra.org/)

Larman,C. and Basili, VR. 2003. Iterative and Incremental Development: A Brief History. *Computer*, 36 (6), 47-56.

Apache Jakarta Project. (http://jakarta.apache.org/)