Implementing Open Source Software for Land Administration Processes in Developing Nations

Geoff HAY and G. Brent HALL, New Zealand

Keywords: Open Source, Land Administration, Processes, Workflow.

SUMMARY

A cadastre is defined by the data items it maintains and the business processes by which those data are modified. Processes in land administration are formal definitions of ordered activities and procedures describing how the state of the cadastre can be modified. To date, efforts to standardise and harmonise land administration (LA) needs towards the development of widely applicable generic software have focused on static data definitions, leaving the process requirements of the domain largely unaddressed. This paper extends our earlier work on developing a dynamic and event-based FLOSS architecture for LA. Our approach defines processes as the key building block for such a system. In the context of a generic LA process, we discuss implementation approaches in terms of the requirements for software that is widely applicable, easily adaptable and deployable in situations that may be less than ideal.

Implementing Open Source Software for Land Administration Processes in Developing Nations

Geoff HAY and G. Brent HALL, New Zealand

1. INTRODUCTION

Less developed and developing nations need software and expertise to implement or upgrade their land records administration systems (Törhönen, 2001; 2004). Proprietary enterprise-level technology and custom-built solutions are more appropriate in situations where there are sufficient resources and expertise to deploy and maintain them. Free/libre open source software (FLOSS) is being investigated as an alternative model for the development of land administration (LA) systems in less well resourced situations (Pieper Espada, 2007; Hall et al., 2008; Hay et al., 2008; Pieper Espada, 2008; Steudler et al., 2010). The primary benefit of this alternative strategy is considered to be the cooperative aspects of FLOSS where jurisdictions may cooperate in the development and deployment of their own solutions.

A cadastre is defined by the data items it maintains and the processes by which those data are modified. To date, efforts to standardize and harmonize LA needs to achieve the development of widely applicable generic software have focused on static data definitions, leaving the process requirements of the domain largely unaddressed. Processes in LA are formal definitions of ordered activities and procedures describing how the state of the cadastre can be changed. This paper extends our earlier work (Hay and Hall, 2009; 2010) on developing a dynamic and event-based FLOSS architecture for LA. Our approach defines processes as the key building blocks for such a system.

For developing nations considering the computerization of their LA systems, an important question is what level of automation is appropriate for computerization of business processes, given that the most widely available resource is human and replacement of human tasks with automation may not be desirable due to costs and other considerations relevant to developing nations. This paper seeks to answer this key question. In Section 2 we define processes as they relate to LA and we describe various implementation approaches. Section 3 describes a generic LA process in terms of sets of human activities and represents a typical LA service performed by humans and with little automation. Section 4 discusses degrees of automation in the context of the needs of developing nations and widely applicable software. The final section provides a discussion and conclusion.

2. PROCESSES IN LAND ADMINISTRATION

Business processes in LA cover a wide range of activities, actors, tasks, and goals associated with the management of land. Processes are established legal procedures that may modify LA data and are generally instigated by and involve interaction with people. Processes can be activities associated with internal management of data about land and its associations, or activities involving external actors and tasks that interact with land management information. A process may be highly automated or may be completely managed and performed by humans, or a mixture of both. While software may be used by humans in performing a specific task (such as digitizing land parcels or processing land registry updates) the software may have no knowledge of other tasks that may need to be performed to achieve the resolution of a process. In less developed nations, processes may involve only human interactions and human tasks with little computer automation. In any case, humans acting as part of a process are expected to understand that process. This understanding may be captured in software code or some formal definition such as a workflow model or activity diagram and these definitions may be simply for human consumption or used to automate a workflow using computer software.

The automation of a process involves encoding the understanding of that process in a way that is executable so that the process tasks can be orchestrated and managed by software to achieve the process goals. Process automation can be achieved with various technologies and with varying degrees of sophistication and complexity. The simplest implementation mirrors a paper-based case management system. That is, database records are created to mirror paper-based records and with the assistance of user interface (UI) forms, officers make entries (dealings) regarding tasks they have performed in relation to a case. Like paper-based systems this kind of system tracks the state of cases and provides some audit and reporting efficiencies over paper-based systems. However, there is little opportunity for automated decision making or integration with tools that officers may use to perform tasks. For example, the task of extracting a parcel record would typically require an officer to invoke separate software to perform a search for the appropriate record and print an extract.

Business process logic (the steps, sequence, and rules) can be encoded directly in software code (hard-coded) allowing for orchestration of tasks in a similar fashion to program ‘wizards’. However, long running processes that endure past software shutdowns and restarts are difficult to manage with such ad-hoc implementation. Such code is also highly specific (to each process) and therefore not easily reusable and is problematic to evolve as needs change. Workflow software decouples the business process logic from software code to resolve these issues.

Business Process Management (BPM) is the discipline of analysing, modelling and improving business processes that result in detailed formal descriptions of processes (usually called workflow models) that may be directly executable. Workflows are composed of various kinds of tasks which might be code modules or tasks to be performed by humans.

Workflow enactment may involve varying degrees of automation. Full automation would involve tasks accessing ‘services’ to complete other tasks. Such services might be provided via Web Services or some other remote invocation mechanism that accesses functionality defined independently of the workflow and that does not require any human intervention. Such functionality might also be implemented in code that is included in the process definition and directly invoked. Partial automation might involve the orchestration of web pages (called page flow) that guide humans through the completion of a task (e.g. data entry or searching). Workflow may involve little or no automation with the workflow composed of fully human tasks that involve simply sending messages to humans (via email or task management software) detailing the task to be performed.

Software implementing an organization's processes must reliably and securely manage changes to the state of the cadastre and its data in a way that is scalable (many actors and many interactions), auditable, and reversible. Workflow engines manage the execution of instances of processes and are responsible for efficiently managing long running processes. This includes maintaining the state of process instances and data that have been entered as part of a process so that process instances can be restarted at any time after deliberate or unexpected shutdown.

Workflow software must also support the management of process logic including activities such as updating the way a process is performed, managing business rules, and the addition of new processes to reflect changing needs. Software should support human tasks and assist human tasking in a way that improves efficiency and integrity. Software should not necessarily impose changes in current practices although it should support the modification and evolution of process definitions as needs require.

Parcel subdivision is an example of a commonly invoked LA process that results in spatial and thematic modifications to a cadastre. The basic process of splitting a spatial polygon (often used in representing land parcels) into multiple polygon parts (reflecting new complete parcels) is a generic and well understood GIS operation and is well supported within generic GIS tools. However, in the context of LA systems it is associated with other operations and tasks which are less generic and which may vary widely between jurisdictions. For example, the identity of parcels must be managed and this includes generating new legal identifiers, managing retirement and creation of spatial objects representing parcels, management of time stamps, the generation of new title documents and other registration tasks, and ensuring that the various rules and legal processes have been observed in the modification of the cadastre. The problem with parcel subdivision therefore relates to the fact that it is typically a long running process with many interactions, actors, and data flows that need to be managed.

3. A GENERIC LAND ADMINISTRATION PROCESS

The following diagrams depict a generic LA process, that is, the kind of steps, activities and actions, decisions, inputs and outputs, and the flow of these steps, that might be expected to occur in the provision of many LA services including parcel subdivision. The generic LA process described here is based on LA processes or services as implemented in two countries, namely Western Samoa and Ghana. Unified Modelling Language (UML) Activity Diagram notation is used to model workflow in terms of human activities and actions (represented by rounded rectangles) in the context of a modest LA office setting where there may be only minimal computerization of the management of a workflow, and where most tasks are performed manually by humans.

The generic LA process is modelled as two consecutive workflows. Figure 1 provides an overview of the basic steps or activities and their order for each workflow. The first workflow (a) depicts an activity servicing a request for information and application documents that are required in the second workflow (b), which is made up of five generic activities. Five of the six activities depicted in Figure 1 are defined in more detail in Figures 2 - 6, which describe generic actions and decisions by officers for each of these activities.

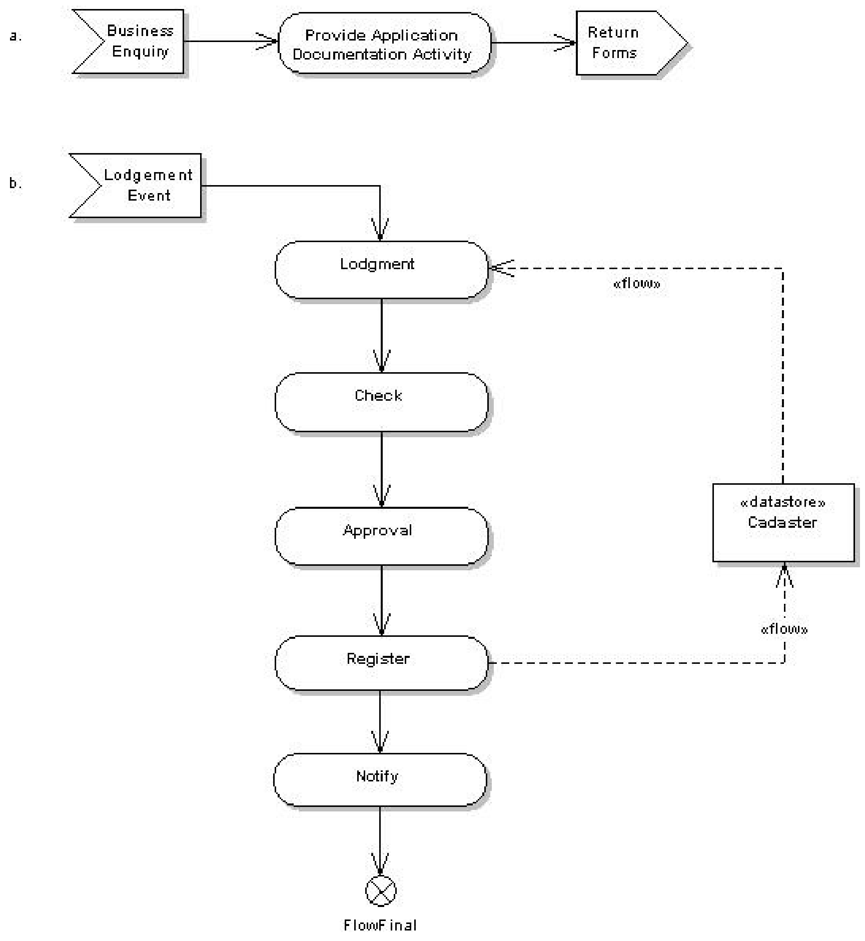

Figure 1. A UML Activity Diagram describing the basic steps or activities composing a generic LA process such as parcel subdivision and/or registration.

The first workflow (a) starts with a business enquiry event, that is, a customer or agent (who might be a landowner, or someone representing and acting for a landowner) makes a request regarding appropriate application documents and/or information regarding the possible lodgement of an application intended to modify the cadastre in some way, for example a parcel subdivision application. The activity named *Provide Application Documentation Activity* represents the activity of LA staff which fulfils the request and results in the return of the appropriate documents to the customer. An officer responding to the enquiry may perform a number of actions to resolve the request and an example of these actions is depicted in Figure 2.

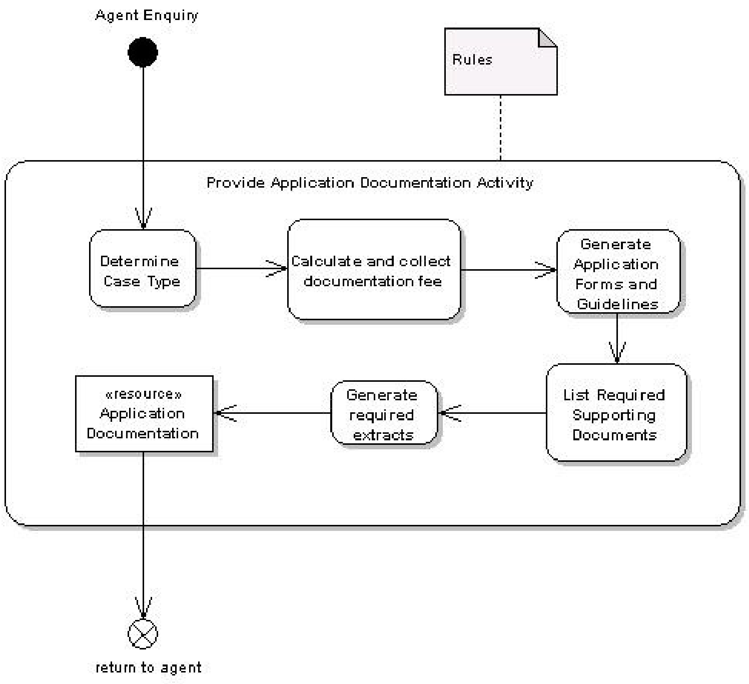
Agent Enquiry
Rules
Provide Application Documentation Activity
Determine Case Type
Calculate and collect documentation fee
Generate Application Forms and Guidelines
List Required Supporting Documents
Generate required extracts
«resource» Application Documentation
return to agent

Figure 2 Activity servicing a request for documentation and application forms.

The *Provide Application Documentation Activity* details actions that are performed by staff to assemble application forms and other documents required by an agent for an application such as a parcel subdivision. The first step is to determine the appropriate case. The officer can then refer to official reference documentation to determine the correct documents and forms. The next action is to calculate and collect the appropriate fee which covers the cost of the documents including printing and other costs that might apply in servicing the request. This fee does not include the fees associated with the lodgement of an application. The next actions involve gathering appropriate supporting documents including guides and any extracts that may be requested. The document set is then combined into a resource pack and returned to the agent to complete the activity. All actions may involve reference to business rules in determining the correct fees and documents.

Such a process can be automated in a number of ways either to assist an officer by providing a software interface that retrieves the appropriate electronic documents based on the type of request, or to replace the officer completely by providing a public Web-based interface that allows agents to order the appropriate forms. In both cases business rules may be used in determining the result. The question as to which approach is considered viable may rely on the availability of Internet and computing infrastructure within a jurisdiction.

After obtaining the set of application forms and other documentation, an agent might then lodge the completed application. The second workflow (Figure 1 b) begins with a lodgement event initiated by an agent. Figure 3 shows the activity as a series of actions that begins with the lodgement event that requires the set of completed application forms as its input. A basic check that all required documents are submitted and completed is performed. The lodgement activity then deals with fees associated with the application, the creation and assignment of a 'case' including a case identifier, the issuance of an application receipt to the agent, and the posting of the successfully lodged case. This is an action that simply represents the completion of the lodgement activity and the passing of the case file to the next activity in the workflow. This action may also represent the point at which the state of the case is saved to a database or filed in some way, and where the activity is signed off as completed. This might also represent the point at which the registry object (e.g. the parcel in the case of a subdivision) is flagged as being currently 'in process'.

The application may be rejected for various reasons and in this case the application is returned to the agent. Upon rejection the case is closed and this is a simplification in order to reduce the complexity in the generic process. An agent may be able to re-lodge the application under the same case but this is not represented here. The black bar in Figure 3 represents the splitting of control flow and hence concurrent execution. In this case there is no need for the application's progress to be halted while the issued receipt is returned to the agent. Most steps in the generic workflow naturally occur in sequence since most actions are performed by humans using paper documents and files which are not easily shared between tasks happening concurrently, i.e. two officers performing different actions with the same case at the same time. More automation of tasks and the use of digital documents implies more opportunity for concurrent actions and therefore more efficient workflow in general.

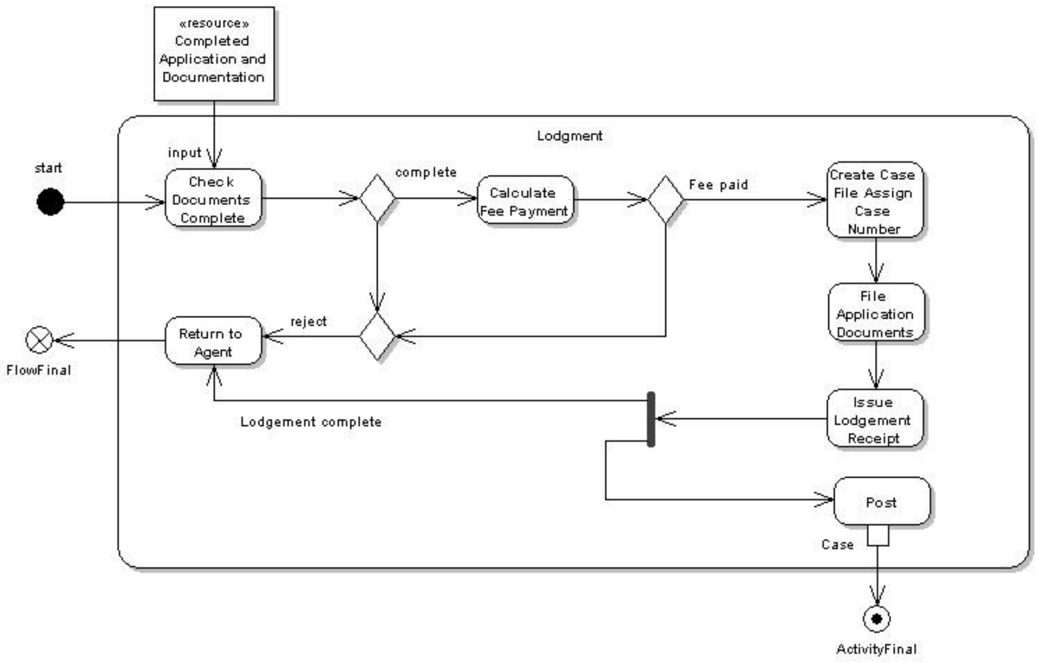

Figure 3 Lodgement Activity from Figure 1 and expanded to show lodgement actions and decisions that may be made by staff. The generic post action represents the successful completion of an activity and the passing of the case file to the next activity.

The *Lodgement activity* is essentially a form filling activity and easily translates to a workstation or a Web-based UI. Typically, a LA office may use paper forms and only a subset of the contained information is entered into a database. Minimally this would simply consist of references to the paper documents or scanned copies which provide little opportunity for automation of tasks. The lodgement activity involves only basic checks that all required documents have been provided and a general check that fields have been filled. It is expected that the lodgement activity would be performed by officers acting in the role of enquiry and customer service staff who may not be qualified to perform more detailed verification and validation checks on the lodged application. These more detailed checks are performed in the next phase of the workflow, the *Check activity*.

Figure 4 shows the actions and decisions making up the *Check activity*. A thorough check of all documents and fields is performed to ensure that all entries are valid and their values are verified. This process may involve referring to other documents such as paper or digital files. Various other checks may be required including spatial aspects such as boundaries and areas, checks to ensure the proposed changes are legal which might include customary considerations, prior restrictions, responsibilities, and so on. Other checks (not shown) might be regarding such things as land use, valuation, zoning etc. Each check involves a decision that may reject the application and return it to the agent. In each check it is expected that notes, comments and perhaps recommendations would be appended to the case file for consideration in the *Approval* phase of the workflow.

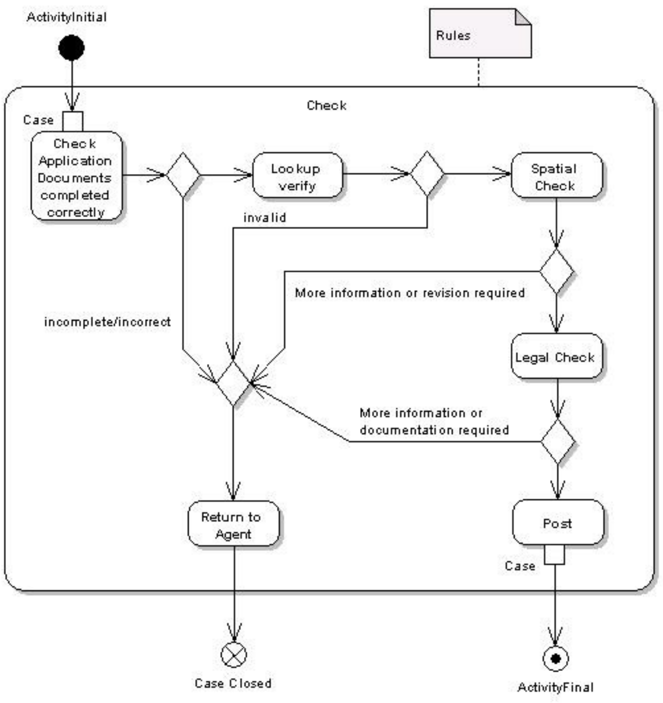

Figure 4 Check Activity. After a case is lodged, submitted documents are checked for consistency, correctness and completeness.

Figure 5 details the *Approval activity* which starts with a thorough review and consideration of any comments and recommendations made in the previous phases. This *Approval activity* would probably be performed by senior officers. A decision is made as to whether the application needs public oversight and this is represented in Figure 5 with the *Publish activity* which represents a public approval process that may result in objections, a legal process and official ruling, and the possibility of the case being rejected. Upon approval of the application, various documents are drafted either electronically or by hand, verified and then signed by the registrar of lands or his/her equivalent.

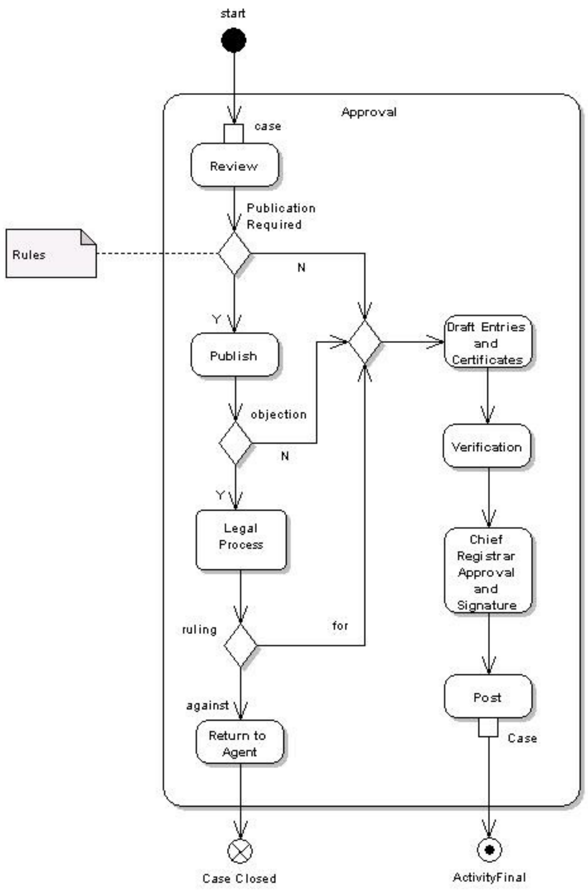

Figure 5 Approval Activity. The approval process may involve a public submission and objection process and an official ruling.

The *Register* activity shown in Figure 6 is responsible for committing the change of state of the official records. The black bars represent a transaction where concurrent operations (which might include insert, delete, and update operations in the database) must be completed before the commit action is performed. After the commit operation, changes are verified and any required documents such as certificates are produced. Ideally, all modifications to the state of the cadastre are done within this single transaction and are fully reversible.

The final *Notify* activity from Figure 1 represents the completion of the process and includes any tasks that need to be performed such as filing relevant documents, notifications, gazetting, and the return of appropriate documents including generated certificates to the agent.

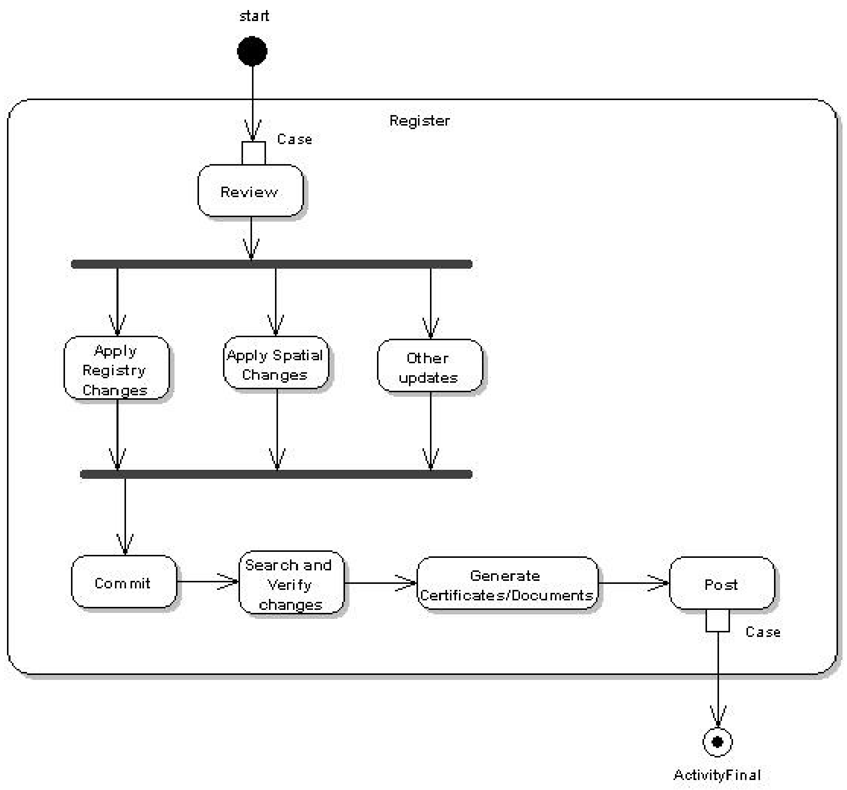

Figure 6 Register. The register involves making appropriate changes to official records and committing these changes.

4. IMPLEMENTATION

The generic processes described previously are composed of human activities and actions. Activities involve human interaction with software programs or paper records to complete a process. The execution of a process for individual cases may be entirely managed by humans and perhaps supported by case management software which replaces paper-based case management. Case management support may simply mirror a paper-based case system using database tables to record cases and dealings. Information collected may consist only of document references, comments, the actioning officer, and timing information for each dealing. Such an implementation relies heavily on the knowledge, expertise and diligence of officers in managing cases and performing tasks, and so is subject to human error that may lead to inconsistent and incorrect information. While such a human-based system may initially be easy to maintain and modify since most changes are simply changing what humans do, it becomes more error prone and change becomes difficult to manage as needs and requirements grow and the number of applications increase. Usually, in computerising a case management system various boolean state attributes are added to the database so that the state of cadastral objects can be maintained during long running processes. For example, a parcel may be flagged with a boolean value indicating that it is currently in the process of being subdivided and is awaiting the result of some other activity. This kind of implementation model (using state based attributes) rapidly becomes unwieldy to maintain as the number of kinds of possible activities and processes increases and the number of attributes required to record state increases.

Workflow management software allows more automation of process management including selecting and assigning tasks to officers, decision making based on input data, rollback and transaction management, management of long running processes, management of interactions with external departments or organisations, and provides a highly sharable formal definition and notation. Workflow software manages the state of records that are ’in process’ and removes this complexity from the database. Hence there is less need for state-based attributes to be maintained within the database. System evolution may be achieved in two ways, namely modification of the workflow, i.e. the order of activities, which is achieved via modification of the workflow definition using appropriate modelling software, and/or by modification or replacement of services that are used by the workflow to complete a task.

The generic process described in Figure 1 can be considered as the orchestration of a set of services. In this sense, each of the activities represents a service that performs a part of the process. From this high-level view, all the services are similar in that they each have one ’start’ and at least one ’end’, and they may specify required inputs or produce outputs. Each action or activity contained within these services may also be viewed as services and with the same general structure as the high-level services. At this level there is no difference between actions that are completely automated and those performed by humans, and there is no specification of the data items and their properties that are to be maintained internally by the service. The notion of generic services affords much opportunity for standardisation and sharing of process definitions.

The high-level activities and actions described in figures 1-6 correspond to human tasks. A simple model is assumed for workflow-human integration where, for each activity or action, a human will be sent a message to perform a specific task (start). There may be relevant documents or references to documents attached to the message that assist in the completion of the task. When the task is completed, the human sends a message to the executing workflow passing on appropriate information (end). This interaction is usually managed by workflow client software (which may be Web-based) that provides logon and authentication for users, and allows users to see all the tasks they have been assigned and to select tasks for actioning. Surveyed workflow products usually include a default client that may be used to manage this tasking.

Some actions may correspond to automated tasks where the workflow executes a code module that performs a specific task (a simple example of such a task could be the creation of a unique case identifier for a new application). Some tasks may be initially performed completely by humans, but later, as the system evolves, human tasks may be partially or completely automated. An example of this could be the Spatial Check activity described in Figure 4 which might initially be performed by hand. For example, an officer loads or enters the plan data into a GIS and executes various operations to check that the plan does not violate any rules (such as minimum size limit for parcels, erroneous leftover area, appropriate setbacks etc). The Spatial Check activity is a good candidate for standardisation and reuse as a set of services for use in such workflow and is easily integrated into an existing workflow.

A number of activities which may be automated and replace human activity require services with knowledge of the specific database and data model, i.e. the tables and fields that are used by the service. An example of this is the Generate Certificates activity from Figure 6 which might simply execute a database report or may involve a complex series of database queries from various sources including spatial queries and operations, or calculations followed by a process that formats and delivers a document for consumption by machines or humans (e.g. machine processible geographic markup language (GML), LandXML, or a formatted Adobe PDF document). Such activities may be highly jurisdiction-specific and are data model-dependant.

Some human activities require the use of a series of UI forms or Web pages that allow officers (or perhaps agents) to enter data, perform queries, and to assist human decision making. These software modules are also highly dependent on the specific data model which might vary significantly between jurisdictions. Standardisation of data models for this domain would allow standardisation of these software modules. However, such standardisation is yet to be achieved. Progress has been made in recent years on frameworks that reduce the development burden for software modules that depend on specific database models. Such frameworks rely on configuration, extension, and code generation via Integrated Development Environments (IDE), and cumbersome middleware to achieve this decoupling. Recent additions to the Java programming language have also been used to reduce further this dependence (for example, the JBoss application server and Seam frameworks utilise the Java Annotation mechanism to decouple database table and field names from the infrastructure code). Complex and highly dynamic and Web-based UIs may be developed with less code with such frameworks, although this requires much more in the way of technical expertise from developers wishing to reuse and adapt FLOSS based on these frameworks.

5. DISCUSSION AND CONCLUSIONS

There is a trade-off between what is practical for a jurisdiction and the needs of community-based shared software. In general, more sharable software implements only generic aspects and relies on configuration and extension to address user needs more completely. These kinds of implementations tend to be more complex than ad-hoc systems which can simply 'hard code' requirements – which makes them less shareable. Ultimately, the measure of what is the correct implementation for this domain (that of LA in developing nations) is that the implementation is effective and is used.

There are two general alternatives to the implementation of processes for this domain. The first is a database state-based case management approach where flag attributes indicate the current state of a process with regards to a data object (such as a parcel), Case management is implemented using database tables which detail duties and responsibilities for officers, and the order of activities to be performed for each case type. Under this process model, an officer can query the database for their current tasks via case management software with completion of tasks resulting in the modification of flag attributes in the database. The second approach uses workflow definitions which are executed using specialised execution environments that manage all aspects of process enactment. Much of the ad-hoc infrastructure code that needs to be maintained in the first approach is provided 'out of the box' in the second approach thereby reducing the initial implementation burden. However, the real advantages of the second approach are in its scalability (many cases) and evolvability (ease of modification as needs change or mature).

As a record keeping system moves from a completely human-managed workflow towards more automated management of human tasks, there are significant benefits to be gained including throughput and service delivery speed, integrity and reliability of data, and significantly improved system evolution. One of the key refinements to a computer solution to LA and records management is the computerisation of long running processes to replace human management of workflows. The benefits of computerising LA systems in general are only realised once a computer system is capable of managing this work. Workflow management systems significantly reduce the burden on officers as well as bringing other data integrity, cost, and time saving benefits. Workflow management systems can manage execution of long running processes and their tasking, and by providing a formal (and sharable) notation a workflow may be defined in diagrams similar to the activity diagrams discussed in this paper. Workflow systems are capable of managing human tasking (i.e. assign tasks to particular people and/or roles) and easily implement much of the generic process described above.

Workflow systems offer much opportunity for standardisation of process definitions across jurisdictions, and allow a clear separation of aspects that may be standardised and those that are more jurisdiction-specific. The notion of code modules as services to workflows provides a clear context in which code may be developed for reuse and sharing, and a useful mechanism by which LA systems may evolve towards more sophisticated automation and expansion of services. For these reasons, it is our view that a FLOSS LA system must incorporate workflow support rather than being based only on ad-hoc user interfaces and state based attributes.

The architecture described thus far is based primarily on workflow software managed human tasking. The next development stage features more complete automation of human tasks and the addition of secure Web-based services for use by external agents (such as lawyers, banks and surveyors) and the general public. Sophisticated service-oriented and Web-based frameworks promise much in the way of integration between internal and external departments and organisations, and integration of disparate specialised software and Web applications. However, this transition brings a dramatic increase in the burden of technology and required expertise to develop and maintain such systems. Such systems also rely on reasonably sophisticated background infrastructure that may not exist in developing nations, e.g. Internet uptake, reliable networks and server hardware, support for online financial transactions etc. In our view the required technology, infrastructure and expertise is beyond that which exists or is likely to exist in the near future in many developing nations.

Our conclusion is that a compromise approach is more appropriate to developing countries. Our approach takes the middle ground and proposes a workflow-based system that automates the management of processes allowing for some standardisation and sharing of process definitions but which is still based primarily on human tasking. Human tasking is integrated with the workflow using client software which already exists in workflow packages but which may be suitable for community development as client software that is more focussed on the needs of this domain. In terms of the integration of more sophisticated workflow services to replace human tasking, our view is that those services that are not suitable for community development are better developed internally by jurisdictions as ad-hoc extensions (which implies limited opportunity for sharing). Moreover, they should be based on existing expertise and infrastructure rather than attempting or immediately requiring the move towards more sophisticated technologies.

ACKNOWLEDGEMENTS

We wish to thank Benjamin Quaye and Pele Fuatai, respectively of the Governments of Ghana and Western Samoa for the facilitating the provision of data for the applied aspects of our research.

REFERENCES

Hall, G.B., Hay, G., Leahy, M., Pieper, G., Goodwin, D., McKinnon, D., 2008. FAO-FLOSS Project - Final Report, FAO.

Hay, G., Hall, G.B., 2009. Architecture for an Open Source Land Administration Application, FIG Working Week, Eilat, Israel, May 3-8, International Federation of Surveyors (FIG),.

Hay, G., Hall, G.B., 2010. A Semantic Web Approach to Application Configuration in the Land Administration Domain, FIG Congress 2010 Facing the Challenges – Building the Capacity, Sydney.

Hay, G., Hall, G.B., Leahy, M.G., Goodwin, D.P., McKinnon, D., Pieper, G., 2008. An open source software solution for land records management in developing nations, GeoCart2008 and 20th Annual SIRC Colloqium, University of Auckland.

Pieper Espada, G., 2007. Open for Change: Scoping Paper on the Use of FLOSS in Cadastre and Land Registration Applications, FAO Land Tenure and Management Unit (NRLA), FIG Commission 7, World Bank Thematic Group on Land Administration, p. %&.

Pieper Espada, G., 2008. Free and Open Source Software for Land Administration Systems: A Hidden Treasure?, Stockholm, Sweden 14-19 June, p. %&.

Steudler, D., Törhönen, M.-P., Pieper, G. (Eds.), 2010. FLOSS Cadastre and Land Registration Opportunities and Risks. Food and Agriculture Organization FAO and the International Federation of Surveyors FIG.

Törhönen, M.-P., 2001. Developing land administration in Cambodia. Computers, Environment and Urban Systems 25, 407-428.

Törhönen, M.-P., 2004. Sustainable land tenure and land registration in developing countries, including a historical comparison with an industrialised country. Computers, Environment and Urban Systems 28, 545-586.

BIOGRAPHICAL NOTES

Geoff Hay

Geoff graduated from the University of Otago with a first class honours degree in Information Science. He has researched in the areas of the Semantic Web, agent-based computing, workflow and process modelling, database design and software metrics. He has been a GIS researcher for a number of years since and has published papers in geospatial aspects of population health. Geoff has recently started a PhD in the area of cadastral information systems with the School of Surveying at the University of Otago, New Zealand.

Dr. Brent Hall

Brent completed his PhD at McMaster University, Canada in 1980. He has held academic appointments in Canada at the University of Guelph, Wilfrid Laurier University, and the University of Waterloo. He has also worked at the University of Auckland and is currently Dean of the School of Surveying at the University of Otago. His area of interest is geographic information systems, in particular Web-based application development and spatial decision support. Brent has co-authored one textbook, edited another, written numerous book chapters, and over sixty papers in peer reviewed journals.

CONTACTS

Mr Geoff Hay
PhD Student
School of Surveying
University of Otago
P.O. Box 56
Dunedin
NEW ZEALAND

Email geoffrey.hay@otago.ac.nz

Dr. G. Brent Hall
Professor and Dean
School of Surveying
University of Otago
P.O. Box 56
Dunedin
NEW ZEALAND

Email brent.hall@otago.ac.nz