A Semantic Web Approach to Application Configuration in the Land Administration Domain

Geoff HAY and G. Brent HALL, New Zealand

Keywords: Open Source, Semantic Web, Temporal Database, Relational Data Model, Object-Oriented.

SUMMARY

This paper discusses the use of Web Ontology Language (OWL) for modeling the land administration domain as an alternative to the use of Object Oriented (OO) approaches such as the Unified Modeling Language (UML). A Semantic Web approach to knowledge representation enables a machine processible domain model to be interpreted at runtime and enhances the prospects for automated decision making. This is in contrast with OO approaches, where software code is generated from the domain model at compile time. The OO modeling paradigm also requires model completeness at compile time and does not allow for runtime emergence or evolution of models. Since knowledge representation models may be interpreted at runtime, the coupling between application and data is reduced and this allows for more adaptive applications to be developed, reducing the need for completely defined and explicit standards. The paper reports on work underway on the development of a layered land administration ontology as part of an effort to produce a general land administration and registry software package that is highly adaptable and applicable across jurisdictions.

A Semantic Web Approach to Application Configuration in the Land Administration Domain

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 suffcient 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; Hay et al., 2008; Pieper Espada, 2008). The primary benefit of this alternative strategy is considered to be the cooperative aspects of FLOSS where jurisdictions may share code and expertise in the development of their own solutions. However, this aspect is somewhat countered by the nature of the land administration (LA) domain where there is considerable variation in the requirements across jurisdictions for compterised solutions thereby making a generic solution possibly less viable.

The Open Source Cadastral Application and Registry (OSCAR) project at the University of Otago (UO) has been investigating the use of FLOSS and open Semantic Web (SW) standards and technology to address this variation in requirements with the goal of providing a more viable, community oriented FLOSS solution. This paper follows from previous work by the authors detailing the architecture and design of UO OSCAR (Hay and Hall, 2009).

The paper reports on progress to date and focuses on the use of the Web Ontology Language (OWL) (www.w3.org/2004/OWL/) for modelling a specific LA domain. Land Registration System (LRS) data were obtained from the Government of Western Samoa for use in the development and testing of the UO OSCAR framework. The second section of this paper provides an overview the general architecture of OSCAR and the benefits of the use of SW technologies. Details of the various existing FLOSS components currently being used in OSCAR are provided. A non-technical discussion of SW technologies and how they are used to form a knowledge base for the OSCAR application is also provided. The third section presents a discussion of the data model and design of the LRS. The third section discusses aspects of the Resource description Framework (RDF) (www.w3.org/RDF/) and OWL modeling, and describes the creation of an OWL ontology for the LRS data and the uploading of these data into the OSCAR system. The final section provides a discussion of the experiences and future work for the OSCAR software and provides details of a preview version of the OSCAR software.

2. OSCAR

2.1 Overview

The OSCAR project was initiated by the Land Tenure Group of the Food and Agriculture Organization (FAO) of the United Nations and the Schol of Surveying at UO to provide a FLOSS package for the administarion of the link between people and land and for use in develoing nations. One of the central issues with this is the variability in requirements across jurisdictions for this type of system. This variability includes such things as spoken language, cultural differences, differences in domain undertanding, and importantly, differences in the links between people and land, and the processes by which these links are managed. This variability makes the provision of a single generic software solution and data model extremely difficult and therefore less viable as a community driven solution. The OSCAR project at UO has been investigating the use of existing FLOSS components and open standards in the development of a general LA software package. In particular, the SW standards (www.semanticweb.org) for sharing and integration of data provide a mechanism for representing knowledge that can be used to reduce coupling between software code and the data being managed. If this can be achieved then the promise is that customization of software becomes more of a configuration task than a coding task. The vision for OSCAR at UO is for a software package that is easily configureable to meet local needs rather than requiring code to be modified and recompiled whenever changes either to workflow tasks or data components are required.

2.2 General Architecture

The UO OSCAR architecture combines a number of FLOSS components and therefore makes full use of existing solutions as well as the communities associated with each FLOSS package. UO OSCAR is an integrating architecture with much use of already proven and tested components. The underlying architecture for this integration is an event-based database model for time varying data and an externalized knowledge model based on SW technologies. The main advantage of this approach is low-coupling between software code and domain understanding. The event-based data model addresses the difficulties associated with maintaining a historical database in a current state based relational model. The externalized knowledge model allows the specific (jurisdiction dependant) data schema and associated rules and understanding to be removed from software code (low-coupling) allowing for cross-jurisdictional differences in requirements to be addressed more independently of software code.

The general architecture overview diagram shown in Figure 1 describes the basic components of the UO OSCAR system including the FLOSS packages that are used for or within each component. Details of each of these FLOSS components are provided below. The architecture defines a data storage and knowledge model component programmed with the Jena Java framework (http://jena.sourceforge.net/) forming a knowledge base that serves as a repository for data and knowledge about the data in the form of a layered ontology. This component uses a standard model for data interchange called the Resource Description

Framework (RDF) which specifically supports the evolution of schemas over time and by implication, the variation of schema across domain space (integration/merging of schema). An external software package named Protégé (http://protege.stanford.edu/) is used for construction and maintenance of the ontology model.

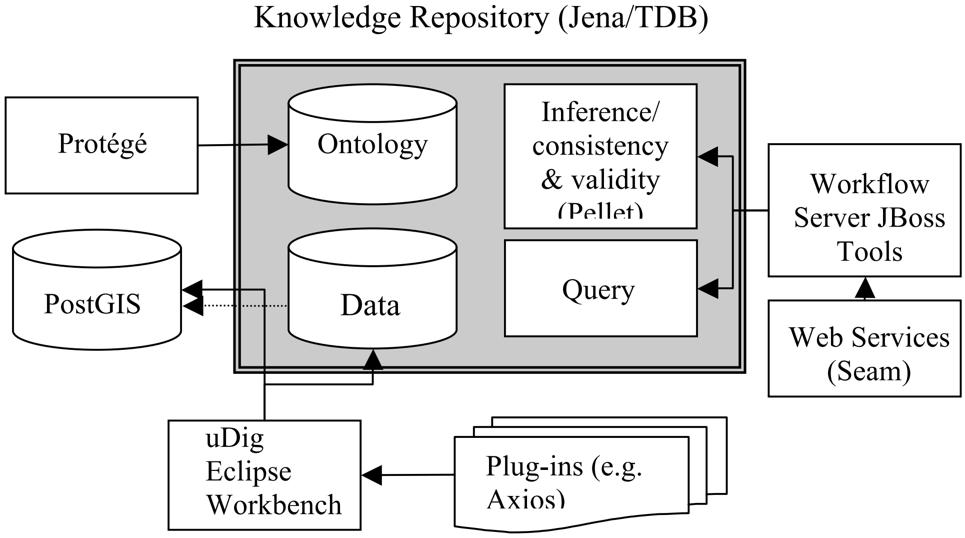

Figure 1: General Architecture Overview

The knowledge base component references a spatial database (PostgreSQL/PostGIS) for the storage of location data such as parcel polygons. The uDig GIS tool (http://udig.refractions.net/) provides an application for manipulating spatial data and this may be extended with various community components (including possibly land registration community components in the future) in the form of Eclipse plug-ins to the uDig environment. The Axios component (see uDig) is one of these plug-ins and provides parcel geometry editing tools such as merge and divide. Eclipse serves as the development environment for uDig and the other OSCAR plug-in components. This provides a useful and highly extensible development environment for OSCAR based LA systems. The plug-in architecture of Eclipse is a useful way of sharing small components for specific tasks. The final component is the orchestration component which essentially serves as a workflow based user interface manager to the LA system. This component provides for the development and execution of orchestrated Web applications for accessing the LA system. The JBoss suite of tools (http://www.jboss.org/tools) is currently under investigation for this aspect of OSCAR.

2.3 FLOSS Components

Table 1 lists the major FLOSS packages that are directly utilized by OU OSCAR. This list does not show those packages that are used indirectly (i.e. those packages which may be used by the main packages). The table provides a description and details of packages including how they are used by OSCAR and the license type associated with each of the packages.

Name	Description	URL	License
Eclipse	Eclipse is an open source community of projects focused on building an extensible development platform, runtimes and application frameworks for building, deploying and managing software.	http://www.eclipse.org/	EPL Eclipse Public License
Protégé	Protégé is a free, open source ontology editor and knowledge-base framework	http://protege.stanford.edu/	Mozilla Public License Version 1.1
Jena	Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. Includes TDB which provides for large scale storage and query of RDF datasets using a pure Java engine.	http://jena.sourceforge.net/	Jena License and Copyright permitting use, modificatoin and redistribution under the same license.
Pellet	Pellet is an OWL 2 reasoner. Pellet provides standard and cutting-edge reasoning services for OWL ontologies.	http://clarkparsia.com/pellet/	AGPL version 3 license.
uDig	The goal of uDig is to provide a complete Java solution for desktop GIS data access, editing, and viewing	http://udig.refractions.net/	LGPL GNU Lesser General Public License
JBoss Tools	JBoss Tools is an umbrella project for a set of Eclipse plugins that supports JBoss and related technology for enterprise systems, for example work flow and web applications.	http://www.jboss.org/tools	LGPL GNU Lesser General Public License
PostgreSQL PostGIS	PostGIS adds support for geographic objects to the PostgreSQL object-relational database. In effect, PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS)	http://postgis.refractions.net/	GNU General Public License

Table 1 FLOSS Components used in OU OSCAR

There is some incompatibility between licenses. Specifically, there is some confusion regarding the coexistence between General Public License (GPL) and Eclipse Public License (EPL) licensed code. Clarification has been sought from the Free Software Foundation Europe in an open letter from the open source software community (Antonello, 2009). Essentially this is a redistribution problem that may result in OSCAR being distributed in a similar way to uDig where there is no “all in one” installer but rather a set of instructions for downloading, installing and integrating the various components. This perhaps makes the installation task a little more cumbersome than would otherwise be the case, but with correct instructions this can proceed without difficulty.

2.4 Knowledge Model

The RDF standard employs a three-part statement (called a triple) comprising a subject, predicate (or property), and object to make assertions about some aspect of the real world. A simple example is the triple: “John” “has father” “Jack”. With the use of Uniform Resource Identifier (URI) schema for naming things, this simple model allows for structured and semi-structured data to be mixed, exposed and shared across applications and domains.

The OWL standard (sometimes called the Open World Language) allows for inference for both consistency (valid according to the ontology) and reasoning (to make implicit knowledge explicit). Concepts about types of things and their associations can be specified in an ontology that possibly links to other shared ontology. An example of making implicit knowledge explicit is given in the simple ontology statement: “has Father” “inverse of” “has Child”. Using this ontology triple together with the previous data triple an inference engine can conclude that “Jack” “has Child” “John”. The following two simple examples show how such ideas might be employed within a land administration domain ontology.

Given the ontology concepts “Agent” (meaning a general class of persons, companies etc that may interact with a land administration system) and a sub class “Land Owner” (meaning an Agent who owns land), the property “owns land” can be said to have a domain of “Land Owner”. An inference engine (under the OWL open world assumption where it is possible to say anything about anything) will conclude that any “Agent” that has a statement with the property “owns land” is of type “Land Owner” even though this fact may never be explicitly stated. Example statements are:

“owns land” “has domain” “Land Owner”
“John” “has Type” “Agent”
“John” “owns land” “some land”

The reasoner would produce the following statement “John” “has Type” “Land Owner”.

The power of this model for integration across disparate data is easily exemplified with the following simple statement to align two ontologies:

“OntologyA-owns land” “same as” “OntologyB-has land owner right”.

Stating the property “owns land” in ontology A is the “same as” the property “has land owner right” in ontology B. A reasoner then may conclude that any subject that has a property “has land owner right” from ontology B is also of type Land Owner from ontology A. Note that the above statements are a simplification designed to convey the general notion and are not necessarily technically correct OWL statements. In RDF, properties themselves may be classified into inheritance relationships and this allows the application software to provide functionality associated with properties that are unknown at compile time.

UO OSCAR uses ontology constructs such as these to externalize as much as possible not only the data but also in the understanding and rules associated with the data, so that this knowledge is not bound up in software code (loose coupling). The advantage of this is that software code need not be modified extensively or recompiled for different jurisdictions. In order for the software to function as a LA application some knowledge of the domain must be built into the code (for example to have different ways of interacting with different types of data). This is achieved with the use of layered ontology shown in Figure 2 where a specific jurisdiction's ontology (such as that for Western Samoa) reuses concepts from a general LA ontology (linking them together), and an application-centred ontology aligns the LA ontology with the application code. Our vision is that the LA ontology is an emergent ontology – as new specific ontology (jurisdictional) are linked into this layered ontology, concepts may be migrated upwards into the LA ontology by community agreement.

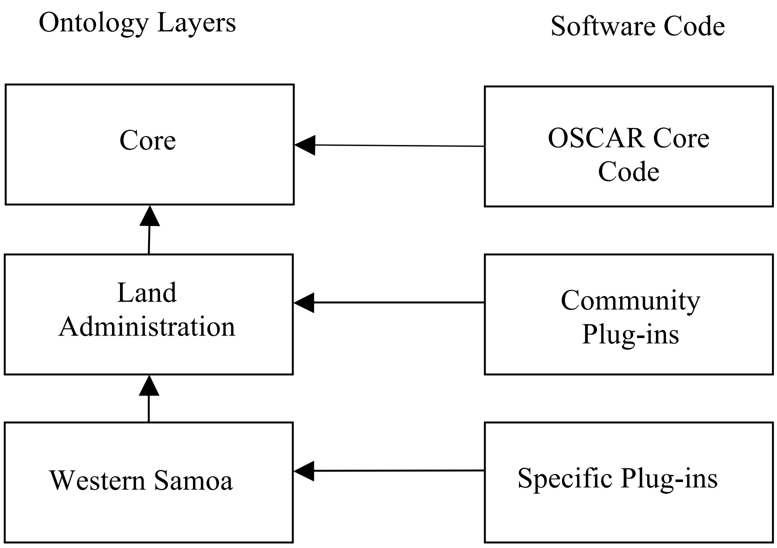

Figure 2: Ontology Layers and Code references

Figure 2 shows three ontology layers including the specific jurisdiction ontology of Western Samoa, the construction of which is described below. The core application ontology forms the link between application code and the ontology data. Concepts in the core ontology are directly referred to in core software code and as such are highly coupled. Terms in this core ontology are, however, very general and are derived from the OSCAR conceptual model which describes basic notions of time, events, basic registry objects (a super class for things that information is stored about), actors or agents (actually a special case of registry object), and amendments which modify states. The OSCAR core software deals in these terms only. The LA ontology specialises core concepts into concepts associated with land administration and this ontology forms the basis for integration across specific jurisdictional ontologies. For example it defines Land Parcel as a subclass of the core Registry Object indicating to the software that a Land Parcel is something that is associated with temporal information. The core software code does not refer to the Land Parcel type explicitly. However, this rule is only relevant to core code, and developers might choose to implement specialized plug-in code that does refer to a specific ontology, and these plug-ins are also shown. Development of these ontologies is ongoing as testing with real data continues. The ontologies are available for download from the UO OSCAR project website (https://source.otago.ac.nz/oscar/OSCAR_Home).

In terms of storage, the knowledge base shown in Figure 1 stores named models containing the data assertions. The named models are supported by the RDF datasets specification (http://www.w3.org/TR/rdf-sparql-query/). The UO OSCAR architecture specifies RDF datasets as groupings of data that logically belong together, such as an instance of a Title which is a dataset containing all the information and links regarding the estate, its shares and references to interest holders. As information is accessed from the knowledge base, combined models are constructed from a series of datasets which are accessed by name and combined into a single model for use by the application. Such a combined model might be the result of a query asking for all the events and associated instruments and estates for a particular parcel. This architecture reduces performance overhead for data access and for any intermediate reasoning or inference that may be required of the software.

The next section describes the construction of an ontology for the Western Samoa land registration data and the upload of the actual data into an OSCAR knowledge base for use in testing. It is important to note clearly that the derived ontology is not an official ontology for Western Samoa land registration. Rather it is used only for illustrative purposes.

3. WESTERN SAMOA LAND REGISTRATION DATA

The current Western Samoa land registration system (LRS) is a custom built database and application software designed to manage the land registry. The system is a traditional data application using a relational data model. The link between the application and the database is via stored procedures. The database and application code are tightly coupled and any changes to the database structure require software code to be modified and recompiled. There are two main parts to the LRS data model, namely the lodgement component which maintains the history of processes that modified the registration data, and the registry component that defines the associations between various agents in the system and land. In terms of the lodgement component, only data describing what happened are stored and there is very little information about the actual rules associated with changes to data – this is coded into the application and/or the database stored procedures called by the application.

As is inherent in the relational approach, the real world understanding is 'normalised' to fit the requirements of the relational model. In uploading the LRS data into the knowledge base this normalization has to be undone and the understanding converted to an ontology for the data. This forms the first of three steps in the process of uploading the data. A brief overview of each step of this process follows.

3.1 Step 1 - Conversion from relational data model to ontology

Types of things, for example types of Instruments, are stored in a lookup table (InstrumentTypeCodeList). This table is easily converted to OWL concepts that are subclasses of the general type Instrument so that each instance of an instrument can be classed as a specific type (for example a Transfer of Title). Tables associated with first class entities in the system are also converted to concepts (for example Parcel, Title). There are several tables in the LRS that store information about agents (LodgingAgent, User, InterestHolder). A general class Agent was defined for all of these data. All table field names were examined and converted to appropriate properties and types. Additional derived statements were also created including such things as defining an inverse relationship between the properties priorTitle and superceedingTitle. An examination of the instance data revealed that some tables (namely Island, LandDistrict, Villages) contain information that is suitable as ontology information rather than data. The distinction between what is terminological data (called TBox data) and assertion data (called ABox data), essentially the distinction between data and data about the data or metadata, is sometimes difficult to determine. Although, for example instances of the islands are not subclasses of island rather they are instances (or individuals) of the ontology class Island. However, by including individual islands in the ontology (TBox) rather than as ABox data, it is possible to express additional rules in the ontology based on the location of instances (for example “a village can have only one particular instance of island which may only be one of the known islands”).

A number of associative entities in the database were collapsed back into many-to-many relationships. This type of entity is only required by the relational model for data. A prime example is the InstrumentTitle table associating many Instruments with many Titles. This association entity does not relate to a specific concept other than perhaps an anonymous container. In the ABox data a title record may have multiple statements with the same property referring to different titles (and vice versa).

3.2 Step 2 – Data upload

Data upload involves creating ABox instances of specific types defined in the TBox and using TBox properties to construct statements linking subjects with their value data. For example, a database record for the person “InterestHolderID=IH123” with the value “John” in the “FirstName” field would result in statements such as:

IH123 is type InterestHolder
IH123 has first name John

The process of creating these statements is highly automated by the Jena framework, so a set of Java programs was constructed to read data from the database and create the appropriate statements stored in a specialized data store. Since RDF data form a directed labeled graph, a graph data store was chosen for storage of TBox data. The Jena triple database (TDB) is a high performance graph store that implements the RDF dataset or named graphs standard that allows separation of ABox assertions into more easily manageable sub graphs.

If a title record has a value for the field “priorTitle”it can be assumed that the prior title has a value for its “superceedingTitle” which references back to the same title. In uploading title records the “superceedingTitle” could be ignored since this information is implied by the “priorTitle” inverse relationship with “superceedingTitle”. A reasoner would be able to infer this information, i.e. make it explicit. This represents a performance trade-off between storage space and processing requirements (reasoners are process intensive). In this case all information contained in the database is made explicit rather than leaving some information to be inferred. The main reason for this is that it allows consistency checking on the actual values in the inverse fields which came from the database.

Conversion of the data model to ontology axioms produced the comparative counts shown in Table 2. Using simple counts of the number of database entities, the number of fields, their types, as well as the associations between tables allows estimation of the number of ontology axioms that must exist to mirror the structure of the database. Counts of the number of atomic statements in the database (i.e. the number of data values in the database which is simply the number of fields multiplied by the number of records for each table) can be compared to the number of ABox assertions. Interestingly, the ABox data contain fewer statements than would be expected and this is mainly due to the large number of null values in the current Samoan database. Null values were ignored during the upload process since nulls do not make sense in a graph architecture, i.e. to say that the property “hasName” has a value of null for the subject John is simply not useful as nothing can be inferred from this statement. Hence, it is exactly the same as having no statement at all, regardless of any meaning we might assign to null.

Total Data Items	1438313
Total Nulls (20%)	294108
Valid Items	1144205
Number of tables	43
Number of fields	196
Number of PK/FK	57
Ontology Statements	3301
Abox statements	874566
Abox named models	73552

Table 2: Comparison Counts

3.3 Step 3 – Ontology alignment

Ontology alignment is the process by which two ontologies are linked together by the production of statements that contain such properties as “equivalent to”, “same as”, “defined by”, “sub class of”, “has type”. Using such statements allowed the test data ontology to be linked with the application ontology (UO OSCAR Core) via the LA ontology. This linking implicitly redefined the basic data model into an event-based model since the application ontology is event-based. The initial conversion of database metadata to ontology data simply mirrored the database types where appropriate.

A number of types in the database are also good candidates for linking with other existing ontologies. A number of ontologies already exist on the Internet and are already shared and reused across domains. Examples are:

- GeoNames, an ontology of place names and general geographic concepts backed by a database containing over eight million place names http://www.geonames.org/
- OWL Time, an ontology of temporal concepts http://www.w3.org/TR/owl-time
- RDF vCard, an encoding that corresponds to the vCard electronic business card profile defined by RFC 2426 http://www.w3.org/TR/vcard-rdf.

This kind of linking between ontologies allows for the sharing of concepts and integration of disparate data, and is the backbone of the Semantic Web. Where possible, links between these foundation ontologies and the test data were created via the application ontology, for example the application ontology contains the concept of latitude and longitude which actually refers to the GeoNames ontology. By linking the test data ontology properties (latitude, longitude fields from the database) to the application ontology the test data are linked to the foundational GeoNames ontology. There are a large number of ontology registry and repositories available on the Internet and these are being investigated for use in this research.

DISCUSSION AND CONCLUSIONS

The UO OSCAR project is an ongoing research effort into the provision of FLOSS software for LA needs in less developed nations. There is no single existing FLOSS package that is able to serve all the needs of in the LA domain or in the diversity of countries that have entered or are entering an era of rapid land title registrations. Given the variability of requirements across these jurisdictions, it is unlikely that a single package and therefore a single community can meet these differing requirements – unless the differences can be harmonized. The research at UO is attempting to harmonize these varying requirements with the use of layered ontology and an event-based model for temporal information allowing the possibility of a single "one size fits all" software application to be considered. To date the research looks promising and it is expected that that by the time this paper is presented, an initial preview version of the software will be available for comment via the project website (www.source.otago.ac.nz).

ACKNOWLEDGEMENTS

We wish to thank the Government of Western Samoa for the provision of real data for our research and specifically, Mr. Vitaoa Pele Fuatai of the Ministry of Natural Resources & Environment, Western Samoa. We also wish to thank Neil Pullar for his help in transferring the data and the provision of various documents describing the Samoan LRS system. We also wish to thank Dr. Mika Törhönen of the Land Tenure Group of the Food and Agricultural Organization (FAO) of the United Nations for his support and belief in the OSCAR project. The FAO provided initial financial support for the project. We wish also to acknowledge the work of Steven Mayhew for his assistance with various aspects of the research under a summer scholarship funded by the University of Otago. Finally we wish to thank Dr. Daniel Steudler for his continued support and invitation to present this paper.

REFERENCES

Antonello, A., 2009. Open letter for request of clarification to the Free Software Foundation Europe regarding the coexistence of GPL and EPL licensed code, http://jgrass.wiki.software.bz.it/jgrass/Open_letter_to_the_fSFEurope.

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. Brent, Leahy, M. G., Goodwin, D. P., McKinnon, D. and G. Pieper, 2008. An open source software solution for land records management in developing nations, GeoCart2008 and 20th Annual SIRC Colloquium, 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.

Pieper Espada, G., 2008. Free and Open Source Software for Land Administration Systems: A Hidden Treasure?, Integrating the Generations FIG Working Week, Stockholm, Sweden 14-19 June.

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 honors degree in Information Science. He has researched in the areas of the Semantic Web, agent-based computing, workflow and process modeling, 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 Hall 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. Dr. Hall has held the Belle van Zuylen Chair at the University of Utrecht in the Netherlands, been a CIES Fellow in Peru, and an Erskine Fellow at the University of Canterbury in New Zealand. His area of interest is geographic information systems, in particular Web-based application development and spatial decision support. Dr. Hall has co-authored one textbook, edited another, written numerous book chapters, and over sixty papers in peer reviewed journals. He is recipient of the Horwood Critique Prize from the Urban and Regional Information Systems Association, of a University-wide award in teaching excellence at the University of Waterloo, and of a national award for excellence in teaching Geography from the Canadian Association of Geographers.

CONTACTS

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

Email geoff.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