South African Journal of Geomatics, Vol. 2, No. 1, February 2013

Talking Titler: Evolutionary and Self-Adaptive Land Tenure Information System Development

Michael Barry, Richard Molero and Abdel-Rahman Muhsen

Geomatics Engineering Department, University of Calgary, Canada, mbarry@ucalgary.ca

Abstract

Conventional land registration systems often do not produce the desired results in uncertain land tenure situations such as peri-urban areas in developing world cities, post-conflict situations, land restitution claims and aboriginal land systems. In the Talking Titler system, flexibility in creating relationships between people and between people and their interests in land has been the primary design feature. It is a tool for prototyping different designs and for developing land tenure information systems usung evolutionary strategies. The methodology was originally conceived in urban informal settlement upgrade projects and land reform and land restitution projects in South Africa in the 1990's. In recent years, the concepts have been tested through interviews with aboriginal peoples groups in Canada, field trials and an initial implementation in land regularization in Nigeria, and a land administration study in Somaliland. The paper overviews the conceptual design of the system, how the design was formulated, testing of the system, and current development. The paper concludes by overviewing an initial design and testing with evolutionary database development and self-adapting software using an extensible markup language (XML) database to reduce the human input into system changes as it evolves.

1. Introduction

Uncertain tenure situations often constitute wicked problem scenarios; defining and agreeing on the problems to be addressed constitutes a major problem in itself (Barry and Fourie 2002a). We present the Talking Titler evolutionary design approach and some of the software design and testing procedures in developing a flexible, evolving, land tenure information system (LTIS) in uncertain situations. We then describe an initial attempt at developing a self-adaptive land tenure information system based on XML data structures and the eXist database as a way to reduce some of the problems associated with evolutionary design approaches.

The perspective is land tenure management in uncertain situations (Barry and Fourie 2002a). The Talking Titler design approach draws on field work in informal settlement upgrades, land reform and land restitution cases in South Africa stretching back to the 1990's, peri-urban customary tenure regimes in Ghana, post-conflict land administration in Somaliland, and land regularisation as part of programmes to mend dysfunctional land administration systems in Nigeria. We have also examined the feasibility of the approach in using recordings of oral history and oral tradition in aboriginal land claims. An initial web based design was presented at the FIG conference in Accra in 2006. Subsequently the design has been developed and tested using MS Access, eXist and PostgreSQL databases (Barry *et al* 2002, Barry 1995, 2009a, 2009b, 2009c, 2010, Barry and Khan 2005, Mason *et al* 1998).

Over the past decade, there has been increased interest in standardized data models for land records. The most prominent is the Land Administration Domain Model (LADM), which has its origins in serving the needs of the European Union and is an ISO land administration standard (ISO/FDIS 19152,). Linked to this is the Social Tenure Domain Model (STDM), which, as with the Talking Titler system, is advocated for situations where conventional registration is deemed unsuited to local circumstances. The STDM adapts the core classes of the LADM, and is under consideration as an ISO standard as a specialisation of the LADM (Lemmen *et al* 2011).

We acknowledge the benefits of ISO standards and standardised approaches to LTIS development if suited to the situation. However, we submit that registering the STDM as an ISO standard is premature given the nature of the situations it is supposed to improve. Firstly, there is a contradiction inherent in applying a standardised solution to problem contexts for which conventional designs (i.e. land registration) are considered inappropriate. Secondly, we submit that an ISO standard stipulating a particular design for these different contexts should emerge after empirical testing over a long period (e.g. 10 years), not before it has been tested thoroughly.

Wicked problem contexts are not well served by grand strategies embodied in a top down design and implementation approach or a standard information system model. From a LTIS design perspective, as a concept, land tenure comprises a set of relationships between people and the land, between different land objects (e.g. parcels, natural features, trees on a parcel, parcels within parcels) and the social capital furthering a group or individual's relationships with land derived from their network of social relations. Even in mildly uncertain situations, several LTIS system starts ups and failures are likely to occur. In a complex situation, it may be difficult to assess whether to place the initial design emphasis on the land objects (e.g. as in most title systems) and lesser emphasis on social relationships, or to place the main emphasis on the changing social network and social capital dynamics and lesser emphasis on the land, or spatial, components. Lastly, we should cater for situations where neither emphasis is suitable, and the land tenure concept expressed above does not describe a particular situation adequately. In these cases, we should allow for a design that emerges and evolve from the data itself, and its properties may be very different to anything observed before.

The thinking has parallels with grounded theory research methodology in the social sciences, arguably the predominant social research methodology nowadays. Grounded theory holds that theory derived from a social study should be grounded in the data from that study alone. It should not be influenced by or compared with existing theory until the final stages of theory development (Glaser and Strauss 1967).

The Talking Titler system can evolve along the lines of conventional land tenure information system designs or emerge as something completely new. First we briefly examine design aspects of the Talking Titler software. We then move onto methods of creating a LTIS using evolutionary methods. Following this we describe a part of our design and experimental work, namely schema evolution and self-adapting software, as a way of addressing the evolutionary system design concept. Finally, we overview an initial attempt at designing and testing schema evolution and self-adapting software methods in land tenure information systems using an XML based database.

2. The Talking Titler Design Philosophy

The Talking Titler conceptual model has four primary classes Media, Person, Land Object and Reference Item (see figure 1). The primary design objective is flexibility so that the system may evolve according to multiple, changing needs in highly uncertain situations. Ease-of-Use and system performance are of secondary importance.

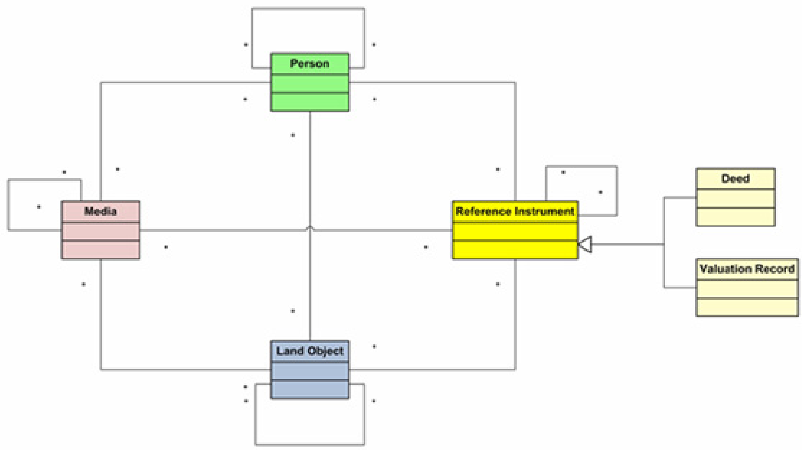

Figure 1 Talking Titler Primary Classes and Specialisations

In reality most LTIS's are driven by a Reference Instrument, such as a title-deed. Local record systems tend to be based on some form of reference document or identifier, such as a local office file number, a shack number, a rent card or occupation permit.

Different types of instruments may be included in the database, providing their type can be ascertained from their unique identifier (i.e. their primary key). As the system evolves, the reference instrument class may evolve into specialisations, such as Deeds and Valuation Records (figure 1). If these evolutions do not work, it should be possible to conflate the specialisations back into the parent Reference Instrument class and start anew, or leave the data in the specialised class dormant, to be accessed if necessary, and let the system evolve into different specialisations. Moreover, if the reference instruments do not work it should be possible to conflate the system to one which models relationships between people and land objects without a reference instrument at all. I.e. we relate people and land objects in the database without using a reference instrument to articulate the relationship. Conversely, we might also start an LTIS by recording the relationships between people and the land without using a reference instrument at all, and the system may evolve to include the Reference Instrument as an additional core class.

As per figure 1, land objects may be related to other land objects (recursive relationships); such as a parcel is a subdivision of a parent parcel or an aboriginal trap line crosses a stream. Similarly, people are related to other people through social networks and social hierarchies; aspects of these relationships provide value (social capital) for a person(s) in asserting (or losing!) interests in particular land objects.

The media class may contain unstructured data items, such as video clips, photographs, sound recordings, hand written notes and reports, and formal or iconic data items such as title-deeds, licences, wills, marriage and divorce certificates, and cadastral survey plans. In concept the Person, Land Object and Reference Item classes may be excluded in the event they do not fit the situation and something very different may evolve. In unusual cases, the design starts with the data, the Media class. It may include one or more of the Person, Land Object and Reference Item classes. It may evolve and then revert to the starting point, i.e. the data in the Media class, if the first design does not work. A possible case is data are collected and related for a land restitution claim. A range of different social relationships need to be captured to determine the rules for membership, to reconstruct a community and settle them on the land. Notwithstanding the influence of social change and local politics, which have a major impact on the success or not of such a project, the more accurate the data and relationships at the outset, the more peaceful the settlement process is likely to be. Over time, the initial restitution LTIS may evolve to form the basis for the tenure rules and general administration once the beneficiaries are on the land, and the original information may prove useful in conflict management (e.g. see Barry 2010).

3. Evolutionary Modelling, Schema Evolution and Self-Adaptive Systems

The evolutionary model is suitable when users do not know what they want initially, but they can formulate an idea about their needs when they see them implemented in a working system. Evolutionary development starts with an initial operational system which gradually evolves over time with continual system changes. The initial system builds the requirements that are well-understood, and it progresses as other requirements become clearer. It involves continual prototyping and development. Prototypes can be throwaway or evolutionary. Throwaway prototypes explore parts of the final system design and are then discarded. Evolutionary forms have many of the features that the software is likely to have in its final form, and so an initial form of the software is gradually developed (Davis 1992, Patel 2009, Budgen 2003, Beynon-Davies *et al* 1999).

Figure 2 portrays an evolutionary LTIS development process flow. It starts with System0, the initial system. Various stakeholders analyse it and request changes which may include changes in the system's purposes and design changes. If feasible the system is changed (System1), the data are migrated, and the cycle starts again. Data migration moves the data from the old design to the evolved one. First data are extracted from the old model, then transformed to the new model's format and requirements, and then loaded into the new system. Finally the data are verified (i.e. error checking and consistency checking) at the end of the migration (Muhsen and Barry 2008).

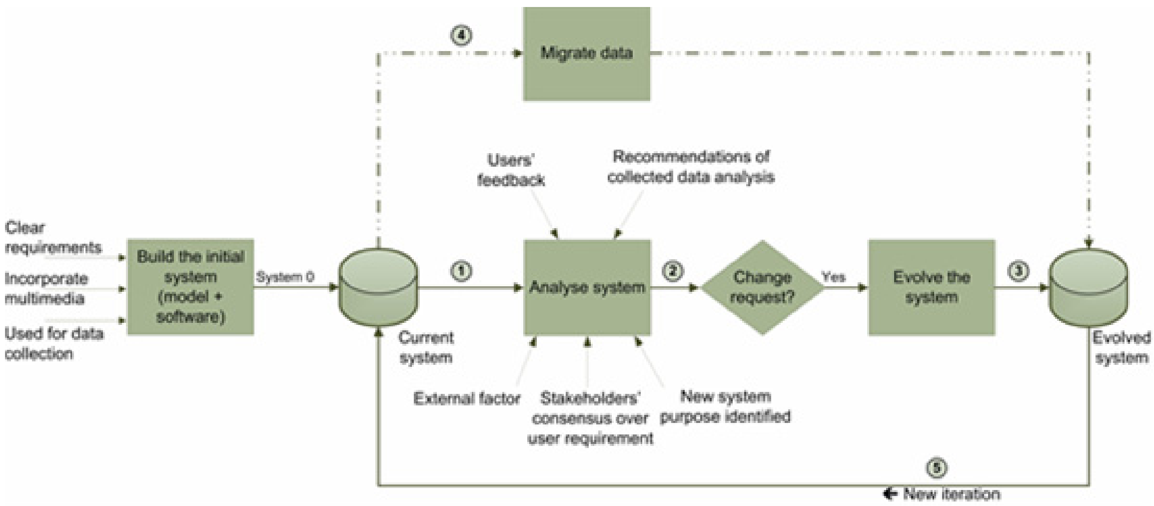

Figure 2 Evolutionary LTIS Development

Ideally, the system should be able to revert back to an earlier state. The system may evolve from System0 through several stages to say System7, for example, and we should be able to revert back to any particular system number. Thus it should be possible to migrate the data backwards, or a copy of the data has to be retained and documented for each system version.

The system may evolve in a number of different ways, and we have experimented with inheritance to create specialisations (Muhsen 2008) and schema evolution (Molero *et al* 2010). Schema evolution implies adding more general classes to the main classes, decomposing the generalized classes into more specialized ones, and adding attributes to these new classes.

One of the many drawbacks of evolutionary development is changes in a system can be frequent, unforeseen, and caused by a diversity of factors. Continual prototyping and development associated with the evolutionary process is expensive and requires skilled personnel (Buckley *et al.* 2005).

Self-adaptive systems may alleviate some of these problems. A self-adaptive system contains built-in support for change, striving to reduce human intervention when a system performs software updates on itself (Salehie and Tahvildari 2009). To achieve the ideal where no human intervention is needed is perhaps utopian, but a self-adaptive approach may enable a significant reduction in human intervention and also reduce the skill levels to change the system.

We now describe the concept of self-adapting software and the use of XML to facilitate the evolutionary process.

4. Self-Adaptive Approach

A self-adaptive system has the capacity to modify itself at runtime in response to changes introduced by an internal decision making process or by external factors such as user input (Andersson *et al*. 2009). A self-adaptive system should be able to handle whole feedback loop iterations of the evolutionary development process on its own (see Figure 2), but this remains an ideal. Drawing on Cheng *et al*. (2009), there are four major factors to be addressed in developing a self-adaptive system:

- Modelling Dimensions. This concerns the definition of models that represent: (1) objectives, (2) changes occurring at runtime, (3) how the system should react to those changes, and (4) dealing with the effects of changes.

- Requirements. This concerns the specification of: (1) what the system should do, (2) what adaptations are possible, (3) constraints on these adaptations, and (4) how to deal with the uncertainty of not knowing what future requirements could be.

- Engineering. This concerns the implementation and control of the feedback loop that underlies the dynamic behaviour of the system which enables internal decision making and self-adaptation.

- Assurances. This concerns the assessment, verification and validation of the changes taking place on the system at runtime.

Addressing the above challenge is not trivial, and there is no general methodology for developing a self-adaptive system.

We perceive a self-adaptive LTIS as flexible; one that allows for the evolution of a basic land administration model into a specialized LTIS suitable for a specific land administration situation. The LTIS components include the user interface, the land records database, and the middleware that allows the interaction between these two. All of these components may be subject to change whenever a change is introduced in the IS, and so they may be affected by the self-adaption process.

To date, we have experimented with self-adaptation at the database level. Database schema evolution is the process through which changes to the database schema are introduced into a populated database without loss of existing data and maintaining database consistency (Roddick 1995).

5. Case Study: Evolution of XML-Based Land Records Database

The advantages of using a XML database are it is text based which makes disaster recovery easier and data are easy to migrate, which in turn means a system can evolve to certain stage, and, once the system needs are clearly understood, data can easily be migrated to a conventional database. We used XML (W3C 2008) which is a widely used mark-up language and format for exchanging data between applications, especially when these applications do not store data using the same structure. Among its benefits are flexibility and ease of use as users can specify their own data structure and extend an existing structure to meet their specific needs (Harold and Means 2002). XML is also part of mainstream computing. Recently the eXist open source XML database was adopted as the data format for government agencies in the USA. All existing data are being migrated to this format to facilitate analysis of data generated by a number of different agencies (Lamont 2009).

We experimented with a self-adaptive design based on a real case of an information system designed to move people from shacks in an informal settlement to formal houses in a newly developed greenfields site (Barry 2006). The study included the design, testing and assessment of a land records database with built-in database schema evolution mechanisms. The prototype used Microsoft Visual Studio C# 2005 code to interact with the eXist-db (v. 1.4) database (Molero *et al.* 2010).

There were four "evolutions" in the experiment.

- System 1 social and demographic information were captured as multimedia and text documents (MEDIA class) and related to the people living in the informal settlement (PEOPLE class) and the shack they occupy (LAND OBJECT class). Thus there were three main classes to start with, MEDIA, PEOPLE and LAND OBJECT (see figure 1).
- In System 2 the local authority decided to relocate the people in each shack to a formal house in a different location. This change is required a new relation between two LAND OBJECTS, the shack and the house.
- In System 3 families are issued temporary occupation permit documents. A new INSTRUMENT class is introduced and related to the existing classes.
- In System 4, formalization of rights over the new land is completed by issuing a title to the house and people are moved from their shack to their house. The title is distinguished from

the temporary instrument through specialization of the instrument class into two different classes.

Changes to the initial land administration model (Phases 2 to 4) involve changes to the database structure which already contains the data collected in the first phase. The introduction of these changes is possible because of the database schema evolution process built in the database. The advantage of using XML as the underlying database model is that it facilitates the introduction of changes to the structure of the database, which facilitates structural changes to the existing data at a later stage.

The prototype architecture is shown in Figure 3. Three main components are: *evolver*, *solver*, and *adapter*. The database is encapsulated (outer dotted line) using two points of entry: as a database administration (DBA) or a regular user (either human or computer). The database has been logically divided into three sections: schemas, transformations, and instances so as to take advantage of collection management capabilities of eXist-db.

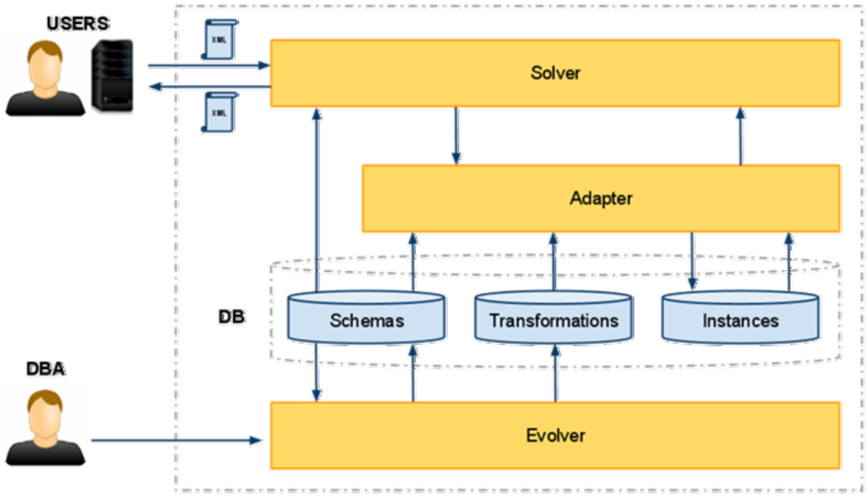

Figure 3 Prototype Architecture

The Database Administrator (DBA) interacts via the *evolver* component, which takes the initial schema and any update primitives that have been registered, and produces a series of transformations (W3C 2007B) that map the previous schema to the new schema. It is important to note that the *evolver* component does not modify any instances; this is because the propagation of changes will take place on-demand when instances are requested as part of a query. The propagation of changes is performed by the *adapter* component.

Regular users interact with the system by performing queries on the data. Queries are first processed by the *solver*, which notifies the *adapter* that instances are being requested by the user.

The *adapter* then applies any pending transformations to these instances, and replaces the query results with the updated instances.

In order to test the prototype, 1000 simulated instances were initially created for PERSON, LAND OBJECT, and MEDIA classes as per phase 1 of the land administration scenario. Relationships were created to represent family ties, and land object occupancy in the initial settlement, and fictitious interviews. As per phase 2, 1000 LAND OBJECTS were created to represent units in the new settlement, and corresponding relationships to the initial land objects were created. As per phase 3, 1000 REFERENCE INSTRUMENT instances were created as were relationships with the new LAND OBJECTS. After schema modifications for each phase transition were introduced, random queries were performed to compare the instances before and after changes to the schema.

The outcome of this case study showed that the complexity introduced by the database schema evolution process can be effectively encapsulated within the database satisfactorily, thus shielding the user from the underlying complexity. However, full automation is not currently possible in the prototype as certain update primitives require further input from the database administrator to guide the change propagation process without affecting the integrity of the database, e.g. the transition from a simple type to a complex type. If there are instances stored using a simple type according to an obsolete schema, there is no way to predict how current values should be converted into new complex values, as a single value may be expected to fill more than one attribute or element. Adaptation in this case requires explicit instructions from the database administrator on how to map the data conversion. Nevertheless, automation of the remaining primitives is expected to reduce complexity and consequently the skill level required for administration of the database, since the operations would be performed without user intervention.

We could not get the self-adaptive process to work entirely with the eXist database, and it has proved to be a complex problem, but we believe that this may be achieved with further work. The self-adapting features of the system performed the database schema changes corresponding to the changes in the model requested by the user, as well as the migration of the data instances affected by the propagation of changes. Updates to the user interface for data collection was adapted internally by the system to allow for collection of data according to the current database schema. The database administrator's intervention was required, however, in complex data migration situations.

6. Concluding Remarks

In conclusion, flexible and evolutionary LTIS are an important strategic option where conventional land registration systems, with their rigid rules and procedures, are not suited to the local circumstances. We have described a conceptual approach to the evolutionary LTIS development problem and one area of design, development and testing that our research group has done in this area. These are concepts that may evolve into practical solutions in the future, but significantly more work needs to be done in this area.

In the meantime, at the practical level, we advocate that flexible evolutionary methods should be developed and applied in computer assisted applications on the ground that accommodate both analogue and digital data. In our observation, the reality is that many digital land information systems are designed and implemented using a linear sequential project based approach, with little attention to the systems required for the system to be a going concern. Thus system planning should accommodate the scenario where a computerised system may collapse completely - for a variety of reasons (e.g. key staff members leave, computer hardware breaks down or is destroyed, failure of management to maintain administrative procedures). An XML based approach to data storage and sharing, along with well-designed methods of storing data outside of the IT system but in parallel with it, e.g. using hard copy plans and documents to perform administration and filing systems that accommodate DVD's and similar devices to store multimedia data, may mitigate some of these risks and facilitate the revival of the system.

The trade-off is that this form of LTIS is it lacks appeal to the user community. As outlined elsewhere (Barry 2009b), the simpler the LTIS, the more likely it will be easy to use and the more likely the system will actually be used. Conversely the simpler the system, the less likely it will provide an adequate model of complex tenure situations or address wicked problem situations, and, in a worst case scenario, it may exacerbate an already troubled situation. The more flexible the system, the more relationships and the greater the level of complexity can be modelled, and the more likely it will be mirror the relevant aspects of local social networks accurately. But, the more flexible and complex the LTIS the less likely it will be easy to use, and the less likely it will actually be used. It is a conundrum that has kept information system designers occupied for a long time, and it should keep LTIS designers similarly occupied.

7. Acknowledgement

This study is supported by the John Holmlund Chair in Land Tenure and Cadastral Systems and the Canadian National Science and Engineering Research Council Talking Titler project.

8. References

Andersson, J, de Lemos, R, Malek, S, and Weyns, D 2009, Reflecting on self-adaptive software systems. In *Software Engineering for Adaptive and Self-Managing Systems, 2009. SEAMS '09. ICSE Workshop on*. Software Engineering for Adaptive and Self-Managing Systems, 2009. SEAMS '09. ICSE Workshop on. pp. 38-47.

Barry, M 1995, Conceptual Design of a Communal Land Registration System for South Africa. *South African Journal of Surveying and Mapping*, December 1995, 23(3), No 139, 153 - 162.

Barry, M 2009, Land Administration Strategy Formulation using GIS in Post Conflict Hargeisa, Somaliland. *Surveys and Land Information Science.* 69(1), March 2009, 39 – 52.

Barry, M, Roux L, Barodien, G and Bishop, I 2002, Video-evidencing and Palmtop Computer Technology to support formalising Land Rights. *Development Southern Africa* 19(2), 261-272.

Barry, M & Fourie, C 2002, Wicked Problems, Soft Systems and Cadastral Systems in Periods of Uncertainty: South African Experience. *Survey Review*. 36(285), 483-496.

Barry, M & Khan, K 2005, *Developing Talking Titler Methods in Creating Land Records.* Research Report to Natural Resources Canada, Surveyor General, Western Region.

Barry, M 2006, Formalising Informal Land Rights: the case of Marconi Beam and Joe Slovo Park. *Habitat International.* 30(2006), 628 – 644.

Barry, M 2009a, Land Administration Strategy Formulation using GIS in Post Conflict Hargeisa, Somaliland. *Surveys and Land Information Science.* 69(1), March 2009, 39 – 52.

Barry, M 2009b, *Standards for Oral Tradition Evidence: Guidelines from First Nations Land Claims in Canada.* 7th FIG Regional Conference, Land Governance and the Environment, Hanoi, Vietnam, 19 -21 October 2009.

Barry, M 2009c, *Innovative Response to Land Registration, Customary Title and Land Conflict: Applying Talking Titler Methodology: Report No 4 February 2009*. British Council, Security Justice and Growth Programme, Nigeria.

Beynon-Davies, PD, Tudhope, D and Mackay, H 1999, Information systems prototyping in practice. *Journal of Information Technology* 14, (1): 107-20.

Buckley, J, Mens, T, Zenger, M, Rachid, A, and Kniesel, G 2005, Towards a taxonomy of software change. *Journal of Software Maintenance and Evolution: Research and Practice*, 17(5), 309–332.

Budgen, D 2003, *Software design*. 2nd Edition ed. Harlow, England: Addison Wesley.

Cheng, BHC, de Lemos, R, Giese, H, Inverardi, P, Magee, J, Andersson, J, Becker, B, Bencomo, N, Brun, Y and Cukic B 2009, Software engineering for self-adaptive systems: A research roadmap. *LNCS*, 5525/2009, 1–26.

Davis, A 1992, Operational prototyping: A new development approach. *Software, IEEE* 9, (5): 70-.

Harold, ER & Means, WS 2002. *XML in a Nutshell*, O'Reilly & Associates, Inc. Sebastopol, CA, USA.

ISO/FDIS 19152. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=51206 viewed 30 October 2012.

Kimball, R & Caserta, J 2004, *The data warehouse ETL toolkit: Practical techniques for extracting, cleaning, conforming, and delivering data.* John Wiley & Sons, Inc.

Lamont, J 2009, Government: leveraging resources for greater effectiveness: innovative ways of managing information can help government agencies achieve their missions while making better use of resources. *KMWorld*, 18(5), 8-10.

Lemmen, C, Uitermark, H, Van Oosterom, P, Zevenbergen J, and Greenway I 2011. The Road to a Standard Land Administration Domain Model, and Beyond. Proceedings FIG Working Week 2011. Marrakech, Morocco.

Mason, S, Barry, M, Crone, S, Geffen, N, and Camroodien, L 1998, Information Management System for Affordable Housing Provision. *South African Journal of Surveying and Mapping*, 24(3), 129-138.

Molero, R, Barry, M, and Hunter, A 2010, Evolution of XML-based land records database for uncertain land administration situations. In: *Proceedings of the Joint FIG Commission 3 and Commission 7 Workshop 2010*, Sofia, Bulgaria, 15-17 November, 2010.

Muhsen, A-R 2008, *Developing Multimedia Land Record Systems*. MSc thesis, University of Calgary, UCGE Report 20283.

Muhsen A-R and Barry, MB 2008. Technical Challenges in Developing Flexible Land Records Software. *Surveys and Land Information Science.* 68(3), 171-181.

Patel, N 2009, 'Theory of Deferred Action: Informing the Design of Information Systems for Complexity', in KD Yogesh, B Lai B, MD Williams, SL Schneberger and M Wade (Eds), *Handbook of Research on Contemporary Theoretical Models in Information Systems*. Information Science Reference, New York.

Roddick, JF 1995, A survey of schema versioning issues for database systems. *Information and Software Technology*, 37(7), 383-393.

Salehie, M & Tahvildari, L 2009. Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4, 14:1–14:42.