OBJECT-ORIENTATION AND INTEGRATION FOR MODELLING WATER RESOURCE SYSTEMS IN SOUTH AFRICA

DJ Clark

PhD LITERATURE REVIEW AND PROPOSAL

Submitted in partial fulfilment of the
requirements for the degree of PhD

School of Bioresources Engineering and Environmental Hydrology
University of KwaZulu-Natal
Pietermaritzburg

April 2011

ABSTRACT

Water is a critical resource in South Africa, which is both a water scarce and a water stressed country. South Africa's National Water Act (Act 36 of 1998) is recognised internationally as being a leader in water legislation, and integrated water resources management (IWRM) is central to the Act's aim to achieve equitable and sustainable use of water resources. To achieve this aim, new and innovative approaches to managing the nation's water resources are required, and this in turn requires innovative and integrated modelling tools to assist water resource managers in understanding and quantifying the water resources under their control. Natural hydrological systems are inherently complex and anthropogenic development in catchments adds to this complexity. The management of water resources in South Africa thus requires water resource modelling tools that are process-oriented, capable of representing real world complexity and suitable for both planning and operations. In this project proposal, water resources modelling is reviewed and two compatible means of addressing complexity in water resources modelling are recommended: (i) object-orientation and (ii) integration of models from different water resource domains. Object-orientation enables model concepts and code to be structured in a more natural and intuitive manner and results in models that are more flexible, extensible and modular. Integration of models developed for specific domains is necessary for IWRM to combine hydrology with ecological, economic, political, social and institutional aspects of water resources management.

TABLE OF CONTENTS

1. INTRODUCTION

Water plays a key role in the social and economic wellbeing of any country (Colvin *et al.*, 2008). South Africa is a water scarce and water stressed country (DWAF, 2004; Colvin *et al.*, 2008). The South African Department of Water Affairs (DWA) reports in DWAF (2004) that 10 of the 19 Water Management Areas (WMAs) in South Africa were estimated to be water stressed in the year 2000, meaning that the demand for water exceeded supply. All, except one, of the 19 WMAs are linked by inter-catchment transfers that assist in the spatial redistribution of water from areas with adequate supply and low demand to highly developed areas with high demand (DWAF, 2004). Colvin *et al.*(2008) make the point that in a water-stressed country, such as South Africa, sound water management is essential. In 1998 a new South African National Water Act (Act 36 of 1998) (NWA, 1998) was legislated for South Africa and Colvin *et al.* (2008) state that South Africa's water laws and policies are "widely recognised as some of the most progressive in the world". Pollard and du Toit (2008) state that South Africa's National Water Act (NWA) requires a different approach to managing the nation's water resources and that integrated water resources management (IWRM) is central to this approach in order to achieve the two main principles of the NWA, namely equity and sustainability.

IWRM recognises that there are hydrological, ecological, economic, political, social and institutional aspects to water resources management. Molina *et al.* (2010) state that the theory of IWRM is widely accepted, but its implementation is difficult as a multidisciplinary management strategy is required that can account for all aspects of water management including economic, social, political, legislative and organizational aspects in addition to scientific hydrological and engineering aspects. Horak *et al.* (2008) state that IWRM requires an integrated analysis of water related issues and a management strategy that reflects the complexity of the water resource system. In this document the term "water resource system" will be used in the context of IWRM, and includes the interrelated hydrological, engineering, ecological, economic and social aspects of water which form the whole system.

The hydrology of catchments is complex even when they are in their natural state and anthropogenic development within these systems adds to the complexity (Kiker *et al.*, 2006). It is widely recognised that water, ecological and social systems are complex in themselves and that the interrelationships and interactions between these systems increases the complexity (Pollard and du

Toit, 2008). Pollard and du Toit (2008) explain that complex systems are distinguished from simple, though potentially complicated, systems by attributes such as non-linearity, uncertainty, scale, self-organisation and feedback loops. The complex nature of water resources systems is largely due to their inherent spatial and temporal variability, and it is important to consider these systems holistically as a hierarchy of interdependent systems and sub-systems (Pollard and du Toit, 2008). Liu and Stewart (2004) state that decision making for natural resource management is complex, requiring communication and consensus amongst the various stakeholders who often have conflicting social, economic and political interests. Pollard and du Toit (2008) conclude that the NWA provides an enabling environment for the management of complex water resource systems.

Modern approaches to water resources management require integrated modelling including all aspects of IWRM (Argent *et al.*, 2000). Typically models are developed for specific domains within the water resource system and integrated water resources management will require integration of the models representing specific domains, for use by multi-disciplinary modelling teams, to provide a systems perspective for water management decisions. Jones *et al.* (2001) state that simulation models provide a means of integrating knowledge from different disciplines and that there is a need for comprehensive integrated models to enable tradeoffs for various policy and management scenarios to be evaluated. Elshorbagy and Ormsbee (2006) state that hydrological models should be integrated with social, economic and management models, or at least provide some means of enabling this type of integration. Growing interest in social and governmental interest in water resources issues in South Africa has resulted in requests by stakeholders for new and better models and modelling tools to assist in the management of the country's water resources (Kiker *et al.*, 2006). These models and modelling tools need to be able to handle and represent the inherent complexity of water resources systems and the need for integrated holistic assessment of water resource systems. Currently in South Africa most water planning exercises by the Department of Water Affairs use the Pitman model to generate monthly streamflows for use in the Water Resources Yield Model (WRYM) or Water Resources Planning Model (WRPM) at relatively large spatial and temporal scales, as described by Mckenzie and Van Rooyen (2003). Dam operations are based on rule curves which are developed to meet a limited number of objectives, whereas there are multiple, often conflicting, objectives related to a range of stakeholders and scales, including many decision variables and constraints (McCartney *et al.*, 2005). A report by Manqoyi and Nyabeze (2006) on dams in South Africa with a capacity of more than one million cubic metres, indicates that operating rules do not exist for many of these dams, many operation rules are based on simple rules of thumb, and that there does not appear to be any use of modelling for dam operations, though the WRYM and

WRPM are often used to develop operating rules. McCartney *et al.* (2005) conclude that the multi-purpose use of dams highlights the need for IWRM and that decision support systems are an important tool to assist decision makers in the planning and operation of dams.

The management of national water resources is important, but complex, and this complexity is recognised by the NWA which advocates IWRM to deal with this complexity, which in turn requires a new way of viewing the management of water resources and requires new modelling tools for decision support. Assessment of complex hydrological systems often requires detailed process-oriented models (Barthel *et al.*, 2006). The object-oriented approach to model design and implementation is ideally suited for use in this type of model and meets the need to better model real-world complexity and the integration of models.

This literature review will start with an introduction to water resource modelling in Chapter 2. Two means of addressing complexity in water resources modelling will then be presented, the object-oriented modelling approach in Chapter 3, and the integration of models from different water resource domains in Chapter 4. Chapter 5 contains a discussion summarising the literature reviewed, followed by some conclusions which will form the motivation for the project proposal presented in Chapter 6, in which changes to the object-oriented design of the *ACRU* agrohydrological model, the design of an object-oriented model input file structure, modifications to the *ACRU* model to make it suitable for operations modelling and the implementation of a means of model integration, are proposed.

2. WATER RESOURCES MODELLING

Water resource modelling is not new, but there are new challenges and approaches to providing the modelling tools required by water resource managers. It is worth pausing to consider what the purpose of water resources modelling is, types of water resource modelling, challenges in modelling water resources and what is required to meet these, and recent developments in modelling.

2.1 The Purpose of Water Resource Modelling

What is modelling and what is its purpose? Modelling is a universal activity, in that every person uses empirical and learned models of the world around them to make everyday decisions (Silvert, 2001). In a professional sense, Silvert (2001) states that modelling is about applying quantitative reasoning to observations about real-world systems to understand them better. Cook and Daniels (1994) state that the purpose of models is to help in understanding the system being modelled, by enabling "What if?" questions to be asked about the modelled system. They explain that well designed object models are versatile, enabling the user to answer questions that were not conceived at the time the model was developed. Argent and Houghton (2001) state that the primary goals of modelling natural environments are to investigate, understand and represent natural systems or phenomena. Simulation models combine scientific knowledge from research with available observed or estimated data to enable estimations to be made for sites where detailed measurements are not available or when conditions at the site change (Kiker *et al.*, 2006). Jones *et al.* (2001) state that simulation models provide a means of integrating knowledge from different disciplines, enable researchers to supplement field experiments with computer experiments, and enable evaluation of tradeoffs for various policy and management scenarios. Javadi *et al.* (2009) make the point that, before developing a model, it is necessary to understand the relevant components of the system being modelled and the interactions between them. Elshorbagy and Ormsbee (2006) state that some of the key steps in modelling are to understand the system being modelled, identify the key variables and represent the physical processes. Silvert (2001) warns that model developers and users need to remember that models are always a simplification of reality, and that any model is only as good as the assumptions, equations and data on which it is based. All models are a simplification of the systems they represent. However, for a model to be useful it must be capable of adequately representing real-world complexity for a given application.

Computer modelling has been widely used for a long period of time in the fields of hydrology and water resources (Argent *et al.*, 2000; Argent and Houghton, 2001; Wang *et al.*, 2005a). Elshorbagy and Ormsbee (2006) state that catchment modelling is a critical part of water resource planning and management as models enable complex hydrological problems to be studied and are being used increasingly to resolve emerging problems and make use of new data sources and techniques. The ability of simulation models to represent locations where detailed measurements are not available or when conditions at the location change, make models useful tools for water policy makers, managers and stakeholders to estimate the impacts of changes in policy, management, land use and climate (Kiker *et al.*, 2006).

2.2 Types of Water Resource Modelling

There are many different types of hydrological model and these are intended to serve different purposes (Silvert, 2001; Elshorbagy and Ormsbee, 2006). There are two broad groups of hydrological models, deterministic models which aim to simulate physical processes, and stochastic models which describe measured time series variables such as rainfall and streamflow in terms of probabilities (Shaw, 1994). Models may also be described as mechanistic, in which the mechanisms of hydrological processes are modelled or empirical where the changes resulting from processes are quantified without representing the mechanisms. Elshorbagy and Ormsbee (2006) also identified two main categories of model used by hydrologists, mechanistic or process-based models, and data-driven models which include regression models, linear and non-linear time series analysis and artificial neural networks. Deterministic hydrological models are necessary due to the practical and financial limitations of measuring hydrological data over long time periods at suitable spatial and temporal scales (Kiker *et al.*, 2006). In a spatial sense, hydrological models may be lumped or distributed, and in a temporal sense may be event based or continuous (Elshorbagy and Ormsbee, 2006). Garrote and Becchi (1997) explain that distributed models describe the spatial heterogeneity of systems by representing the characteristics and processes of spatially heterogenous phenomena in a local manner. Garrote and Becchi (1997) conclude that in most cases it is better to use simple equations together with good spatially detailed data, than to use complex equations with lumped information that does not represent spatial heterogeneity. Wang *et al.* (2005a) state that complexity is one of the main concerns in catchment modelling, where fully distributed models are able to represent variations in space and time, but have extensive requirements for data and expertise, while lumped models are easier to use but are generally not able to represent interactions between the spatial entities within the system being modelled. The water resources modelling discussed in this review is focussed on deterministic, distributed type models which may further be mechanistic or empirical, or a combination of these.

2.3 Challenges in Water Resource Modelling

One of the challenges of environmental modelling is that it includes phenomena that are spatially distributed and temporally dynamic (Bian, 2007). Catchments display a wide range of heterogeneity in their spatial attributes and complexity in response to climate inputs that vary both spatially and with time (McDonnell *et al.*, 2007). Some catchment attributes vary continuously in space, such as altitude and soil type, while others vary continuously in time, such as streamflow (Kumar *et al.*, 2010). Wang *et al.* (2005a) observe that reductionism is often used in hydrological studies as it highlights individual processes, but that a wider view of the whole system requires that the interactions between processes should also be considered. Despite the early application of computer modelling in the field of water resources there are still many limitations. Wang *et al.* (2005a) make the point that though water resource and environmental models have been widely applied to a range of problems, they are still simplified representations of hydrological or ecological processes being modelled. Wang *et al.* (2005a), citing Wang *et al.* (2000), state that there are two primary reasons for this simplification: (i) our limited understanding of the physical processes being modelled and (ii) the inability of a model to represent all known phenomena. Bian (2007) points out that data measurements are discrete measurements in time and space of continuous phenomena, but that these are sufficient for modelling as the objective is to represent the real-world and not to replicate it. Wang *et al.* (2005a) conclude that a model is no better than the understanding of the system being modelled, and that most existing hydrological models are simplified representations of specific, selected processes to solve specific research questions. Argent *et al.* (2000) and Argent and Houghton (2001) agree, stating that the early and continued adoption of computer modelling in hydrology has resulted in a legacy of models, many based on obsolete software engineering practices, which address similar problems, though there are also many models that have been developed to study specific problems or locations. Elshorbagy and Ormsbee (2006) make the point that mechanistic models are usually data intensive and that the application of these models is dependent on data availability at the study site.

2.4 Water Resource Modelling Requirements

Though there are currently still many limitations in modelling water resources modelling, the requirements are clearly recognised in the literature. Wang *et al.* (2005a) state that simulation is a key part in the hydrological assessment of catchments, and that it is important that the model selected be able to adequately represent the behaviour of the catchment being studied. Elshorbagy and Ormsbee (2006) state that there is a need for hydrological simulation models that are able to represent complex hydrological systems in a realistic manner. A useful set of required characteristics for hydrological simulation models were identified by Elshorbagy and Ormsbee (2006) as follows: (i) describe and simulate the hydrological system in a simple manner, yet represent complex hydrological systems in a realistic manner, (ii) support simple and advanced modelling options based on data availability, (iii) be able to represent the dynamic nature of hydrological systems, (iv) be able to represent linear and non-linear processes, (v) be able to represent feedbacks, (vi) be able to represent natural and human induced characteristics in the system being modelled, and (vii) enable management and policy scenarios to be evaluated. Alfredsen and Saether (2000) mention the need to separate calculations representing physical processes from the model components that describe the system being modelled. Kokkonen *et al.* (2003) explain that environmental problems are typically multi-disciplinary, require analysis of data from several sources and the application of simulation and analysis tools developed by experts in different disciplines, and that integration of datasets and modelling tools is a challenge for integrated water resources management. Jones *et al.* (2001) states that an important objective when structuring a model, should be to enable researchers from different disciplines to contribute to scientific aspects of the model without needing to be concerned with the non-science aspects of model construction. According to Garrote and Becchi (1997), the processes in distributed hydrological models are data intensive, and thus data storage, access, and management are an important part of modelling. Alfredsen and Saether (2000) also point out that hydrology and hydraulics models are data intensive and thus require an efficient means of handling data. Garrote and Becchi (1997) emphasise that models should be flexible with regard to their data structures and process representation to adapt to different levels of data availability. Argent *et al.* (2000) and Argent and Houghton (2001) state that modern approaches to catchment management require integrated modelling, including hydrological, ecological, economic and social aspects. They state further that complexities in catchment modelling, due to non-linearity in many natural systems and the need to model across spatial and temporal scales, requires a flexible modelling approach.

2.5 Recent Developments in Water Resource Modelling

There are many examples in recent literature of new approaches to water resources modelling and there appear to be two main development thrusts, one being better representation of real-world entities, interrelationships and processes, and the other being integration of discipline specific models. Argent *et al.* (2002) note that developers of environmental models are realising the benefits of adopting modern approaches to software engineering. Based on the requirements for hydrological modelling identified, Elshorbagy and Ormsbee (2006) conclude that an object-oriented modelling approach will enable simulation models with the above characteristics to be developed, and that it is easy to extend these models as understanding of the modelled system develops. Elshorbagy and Ormsbee (2006) explain that the object-oriented approach requires the complex relationships between different entities within a system to be understood and results in a simulation model that can represent and quantify the behaviour of the system.

The object-oriented approach to modelling has been widely applied in the fields of hydrology and ecology and strong support for this approach can be found in literature. Spanou and Chen (2000) state that advances in computer science and computational techniques during the past few decades, have contributed greatly to the assessment and management of water resources, and in dealing with their complexity, and that object-orientation is at the forefront of these developments. Javadi *et al.* (2009) describe simulation as a practical and effective technique for analysis and problem solving, which has been applied in water resources for several decades and is facilitated by recent advances in computer programming techniques, particularly object-orientation. Hydrological systems are inherently complex and abstract representation of these systems in computer code to create models can add even further complexity (Kiker *et al.*, 2006). Object-orientation is one approach to modelling that tries to manage this complexity by representing such systems in a more intuitive manner (Band *et al.*, 2000; Clark *et al.*, 2001; Kiker and Clark, 2001b; Wang *et al.*, 2005b; Kiker *et al.*, 2006).

3. OBJECT-ORIENTED MODELLING

The world can, to a large extent, be viewed as being composed of discrete, physical, interrelated objects. Each of these objects has three main characteristics: identity, state and behaviour. Some objects may be simple, while others are complex and are composed of a number of smaller constituent objects. The question to be addressed is how can this object-oriented view be replicated in computer models, especially in water resource system models?

Object-orientation is based on the perception that the real-world consists of objects which interact, and is a means of structuring models using objects as humans perceive these objects in reality (Egenhofer and Frank, 1992). Bian (2007) states that "The fundamental intention of the object-oriented paradigm is to represent human perceptions of the world.". Object-orientation represents a change in computer modelling away from a computer-oriented or solutions-oriented view to a knowledge-oriented view based on how the real-world is perceived (Bian, 2007).

Object-oriented design and modelling may be described as an approach to thinking about problems using models based on real-world concepts using collections of discrete objects that include both data and behaviour (Rumbaugh *et al.*, 1991). Bian (2003) states that object-orientation is intended to provide a natural representation of the real-world, on the assumption that the real-world is composed of a hierarchy of interrelated objects each containing attributes and behaviour. Cook and Daniels (1994) explain that just as ecology deals with organisms and their relationships to each other and their surroundings, object-orientation deals with objects and their relationships to each other and their surroundings. According to Gärtner *et al.* (2001), object-based models are based on the concept of objects consisting of attributes, behavior, and relationships with other objects in space and time.

Wegner (1990) documents that the object-oriented paradigm emerged in the 1980s and is the result of an evolution in programming languages starting with assembly languages in the 1950s, followed by procedural languages in the 1960s, then structured programming and data abstraction in the 1970s. Wegner (1990) points out the versatility of the object-oriented paradigm which "supports multiple but coordinated paradigms of thinking and problem solving", and makes the following statement about the object-oriented paradigm, "Its universality as a robust representation, modelling, and abstraction technique suggests that the object-oriented paradigm is conceptually and computationally fundamental.". Wang *et al.* (2005b) state that object-oriented design and programming provides a different means of organising models, with organisation around classes or objects containing both attributes and behaviour, and the emphasis is on the design rather than the coding details. The object-oriented approach focuses on the problem and structure, rather than function or solutions and that full use of the object-oriented approach requires technical and organisational changes, beyond just using an object-oriented programming language (Cook and Daniels, 1994).

There are two roles for object-orientation in modelling: (i) as a means of representing the real-world system being modelled to promote understanding, and (ii) as a programming technique for implementation in computer code (Rumbaugh *et al.*, 1991; Bian, 2003). Lal *et al.* (2005) state that previously object-oriented design may not have been considered important to hydrologists, but that increased complexity in modelling hydrological process and the need for integration across disciplines is changing this view. Object-oriented design techniques and object-oriented programming languages with which they may be implemented have provided an opportunity to explore new methods of representing complex hydrological phenomena (Wang *et al.*, 2005b).

In the literature the terms object-oriented analysis (OOA), design (OOD) and programming (OOP) are used. Bian (2003) and Bian (2007) describe these as being three levels of abstraction, where OOA is the development of a conceptual model of real-world objects, relationships and events, OOD is the definition of a formal model of these real-world objects, relationships and events, and OOP is the code implementation of the OOD in a programming language. These levels are simplified by Cook and Daniels (1994) as three viewpoints: (i) that of an observer of the real-world, (ii) that of a software specifier and (iii) that of a software implementer. Bian (2007) states that the conceptual OOA and OOD levels should remain independent of implementation, as implementation may change with the computing environment.

3.1 Basic Concepts of Object-Orientation

A few of the basic concepts of object-orientation including objects, encapsulation, classes, inheritance, aggregation, association and polymorphism are explained briefly in this section. Detailed explanations of the concepts on which object-orientation is based and the application of these concepts can be found in books such as those by Rumbaugh *et al.* (1991) and Booch (1994).

An object is defined by Rumbaugh *et al.* (1991) as “a concept, abstraction, or thing with crisp boundaries and meaning for the problem at hand”. Objects have identity, attributes which describe their state, and behaviour which can change the state of the object (Bian, 2003; Bian, 2007). Objects may be described as programming units that combine data, as instance variables, and operations or functions that use or modify that data, as methods (Wegner, 1990; Jones *et al.*, 2001; Wang *et al.*, 2005b).

Neither real-world or software objects can be uniquely identified by their attributes or behaviour, and thus require a unique identity in order to be distinguishable and persistent (Wegner, 1990; Rumbaugh *et al.*, 1991). Rumbaugh *et al.* (1991) believe that the identification of the objects within a particular problem domain depends on the nature of the problem and the judgment of the modeller. Bian (2007) states that object-orientation is intended as a generic model for use in all disciplines and does not provide any clear principles of how objects and their attributes, behaviour and relationships should be defined. Each discipline may choose it own interpretation and definition of objects, and Bian (2007) notes that there is a danger that object-orientation may be applied inappropriately to non-discrete phenomena. Bian (2007) suggest five criteria for identifying spatial entities: (i) spatial scale, (ii) the existence of a boundary, (iii) a common set of attributes, (iv) common behaviour or processes, and (v) type of mobility. Bian (2007) then goes on to differentiate between spatial objects, which have clearly defined physical boundaries, and spatial regions which are a portion of continuous space with definable but non-physical boundaries, where both spatial objects and spatial regions may be represented by software objects.

Bian (2003) and Bian (2007) explain that the concept of encapsulation refers to the containment of identity, attributes and behaviour within objects. Encapsulation provides the ability to hide some internal details of an object, and expose only those details required for interfacing with other objects and manipulation of hidden attributes in a controlled manner (Silvert, 1993; Molina *et al.*, 2010). Silvert (1993) states that encapsulation, through which the internal variables in a class are only accessible through methods belonging to the class, may at first appear to be a hindrance, but is important for the efficient use of object-oriented programming.

Objects of the same type are represented by an abstract class (Bian, 2003; Bian, 2007). Rumbaugh *et al.* (1991) explains that objects with the same attributes and behaviour can be represented by a common class, which is an abstraction of the objects it represents. Wang *et al.* (2005a) explain that in object-orientation a class represents a blueprint or description of a collection of similar objects, where objects are described as being instances of a particular class. Wegner (1990) describes classes as being templates from which objects can be created. Silvert (1993) describes the concept of classes, which are important in object-oriented programming, as one of the most intuitive and useful characteristics of this approach.

Molina *et al.* (2010) describe polymorphism as enabling objects to have different natures. Rumbaugh *et al.* (1991) explain that polymorphism means that the same operation or type of behaviour may behave differently depending on the type of class being acted on or the type of class performing the action. Silvert (1993) explain that polymorphism as being the mechanism by which a single method can have different implementations within a class or sub-classes depending on the information passed to it.

Inheritance is the sharing of attributes and behaviour between classes based on a hierarchical relationship (Rumbaugh *et al.*, 1991). Inheritance enables classes to be organised into a hierarchy where subclasses inherit attributes and behaviour from their parent class, known as their super-class, helping to reduce redundancy (Egenhofer and Frank, 1992; Bian, 2003; Bian, 2007; Molina *et al.*, 2010). Generalisation occurs as one moves up the class inheritance hierarchy, and specialisation as one moves down the class hierarchy (Egenhofer and Frank, 1992; Gärtner *et al.*, 2001). Kiker *et al.* (2006) explain that inheritance enables the creation of powerful classes at the top of the hierarchy which may be inherited by simpler classes lower down the hierarchy. Single inheritance is a simplification as the real-world as it is strictly hierarchical, where each class has only one immediate super-class and belongs to only one hierarchy, however multiple inheritance is difficult to implement and use, as complex rules are required to resolve clashes in instances where attributes or behaviour with the same name are inherited from both super-classes (Egenhofer and Frank, 1992). Inheritance is a powerful means of generalising models, and the ability to reuse and modify code through inheritance is one of main advantages of object-oriented modelling (Silvert, 1993). Inheritance enables "type of", "kind of" or "is a" relationships between objects to be specified.

The concept of composition refers to the description of the way in which objects are related to each other, or organised, through inheritance, aggregation and association (Bian, 2003; Bian, 2007). Aggregation, refers to the fact that an object can consist of other sub-objects, in other words, objects may be composite (Egenhofer and Frank, 1992; Bian, 2003; Bian, 2007). Aggregation results in "part of" and "consists" of relationships. Association refers to the existence of a relationship between two or more independent objects (Egenhofer and Frank, 1992). Association enables the specification of specialised relationships between objects that cannot be represented by inheritance or aggregation (Bian, 2003; Bian, 2007).

3.2 Advantages and Disadvantages of the Object-Oriented Approach

There are many legacy models that were developed prior to the widespread adoption of the object-oriented approach to model design and programming that are still in use. There are, however, many advantages to the object-oriented approach, both in terms of representation of the system being modelled and as a method of programming.

Wirth (2006) mentions that object-orientation had its origins in the field of system simulation and states that as this paradigm closely reflects the structure of real-world systems it is well suited to modelling complex systems with complex behaviour. Silvert (1993) draws parallels between the structures of object-oriented models and the ecosystems being modelled, and found a strong resemblance, which lead to the suggestion that the structure of a model should reflect the structure of the system being modelled. Reitsma and Carron (1997) state that object-oriented programming has facilitated the types of model where river and reservoir systems are represented as networks of discrete entities, and provides a data model in which there is a close correspondence with discrete physical entities or discrete processes. They explain that though objects behave in a largely autonomous manner, their collective behaviour represents the behaviour of the system as whole. Lafore (2002) state that real-world entities can be more accurately represented by programming objects that combine data and functions. Wang *et al.* (2005a) conclude that the object-oriented approach is a more natural and intuitive means of modelling real-world objects, through their combined attributes and behaviours. They also conclude that object-oriented design and the concept of classes is a powerful conceptual tool for describing complex hydrological processes. Gärtner *et al.* (2001) state that though model development requires detailed knowledge of the domain being represented, object-orientation is a modelling tool that promotes understanding of the domain.

Bian (2003) notes that conceptually object-orientation is based on the assumption that the real-world consists of discrete objects, however, continuous phenomena, such as space or the environment in which organisms exist, and other natural phenomena, are often conceptually continuous. Bian (2003) warns that this conceptual mismatch may lead to difficulties when formalising representations of the environment using the object-oriented approach.

Martinez *et al.* (2008) state that the object-oriented concepts of inheritance, aggregation and association enable the development of models that are modular, extensible and flexible. Kiker *et al.* (2006) believe that models developed using the object-oriented approach are inherently modular, which Silvert (1993) states is an advantage to modellers as it enables groups of scientists and programmers to work on separate modules within a model. One of the main advantages claimed for the object-oriented approach is its extensibility and associated reuse of code through inheritance (Silvert, 1993; Wang *et al.*, 2005a). Sydelko *et al.* (2001) make the point that inheritance makes object-oriented models easier to modify, as new specialised objects can be created that inherit their primary features from existing objects, and can be used in existing modules without necessarily requiring changes to these modules. Martinez *et al.* (2008) found that the object-oriented concept of inheritance enables new modelling approaches to be included in a model, by extending the model, thus reducing effort and leaving existing model functionality intact. Similarly Bian (2003) states that the extensibility of object-oriented models, provided through inheritance and polymorphism, enable a model to be extended without changes to the original code. According to Martinez *et al.* (2008), the object-oriented approach enables a model to be easily tailored to conditions at a specific site, without conflicting with existing model code. Another advantage often claimed for the object-oriented approach is its flexibility. Sydelko *et al.* (1999) state that an object-oriented architecture is more flexible and modular. Wang *et al.* (2005a) state that using the object-oriented approach they were able to constrain complexity by increasing simulation flexibility. Shane *et al.* (1996) state that object-oriented programming enables the creation of software models that are easier to use across computer platforms and which can be tailored to the specific requirements of model users.

The object-oriented approach is often compared to the procedural approach previously used for developing models. Lafore (2002) mentions several advantages of an object-oriented programming language over procedural languages, including: encapsulation which enables objects to prevent or intercept errors, and simplify writing, debugging and maintenance of code, inheritance which helps reduce code length by reducing duplication, and making the existing relationships between parts of a model clearer, and also makes models more flexible through increased extensibility and reuse of code, and polymorphism which simplifies the design of models and can lead to more efficient programming. Gärtner *et al.* (2001) state that an advantage of the object-oriented approach to modelling is that objects encapsulate state using attributes, behaviour using methods, and some means of communicating with other objects. Wegner (1990) explains that models using the procedural approach are a sequence of actions, while object-oriented models consist of collections of related components in which interaction between objects in the application domain is modelled directly. Silvert (1993) states that the main concept of object-oriented programming is to represent the interaction between abstract code representations of real-world entities, compared to procedural programming which consists of a linear sequence of calculations. Silvert (1993) describes procedural simulation models as consisting of sets of variables and instructions, where the list of instructions describes the manner in which variables values change with time. Silvert (1993) describes object-oriented simulation models as consisting of collections of objects, each object containing complex internal dynamics, and where the objects interact with each other based on the properties of the interacting objects. Wang *et al.* (2005a) state that object-oriented design facilitates easy model extension, and enables processes to be described in a "natural, direct, concise, and adaptable manner" compared to models created with a traditional procedural design. Wang *et al.* (2005a) and Wang *et al.* (2005b) describe the procedural approach as being action oriented, with the focus being on the function, and where the function and the data are separated. They argue that complex real-world entities such as catchments and the entities, such as rivers or soils, of which they are composed, contain both attributes and behaviour which means that they cannot be represented completely by just data or functions. Kiker *et al.* (2006) explain that hydrological models evolve over time due to changing needs and better understanding of hydrological processes, and in the procedural approach existing model code would need to be modified, while in the object-oriented approach models can be extended through inheritance using new code and generally leaving existing code intact. Wang *et al.* (2005b) acknowledge that model users may not be able to differentiate between models designed and implemented using the object-oriented or procedural approaches, but that model designers will realise the advantages of code reusability and maintainability with the object-oriented approach.

Simonovic *et al.* (1997) describe the object-oriented modelling approach as being different to the function-oriented modelling approaches in which the focus was on identifying and decomposing system functionality. They explain that, the object-oriented approach focuses on identifying the objects making up the application domain, then fitting the functionality around them. They go on to state that object-oriented models are more robust as they evolve, as they are based on the underlying application domain as opposed to the functional requirements of a specific problem.

3.3 Suitability for Water Resources Modelling

Leone and Chen (2007) state that object-oriented technologies are forming an increasingly important part of software development for water resource management. Alfredsen and Saether (2000) believe that hydrological systems have an object-oriented structure, and that given the increasing requirement for integration of models within hydro-informatic systems, object-oriented modelling should provide clear advantages in the field of hydrological modelling. Wang *et al.* (2005a) note that the adoption of object-oriented modelling techniques requires a paradigm shift in the manner hydrological processes or events are conceptualised, compared to traditional procedural approaches. Simonovic *et al.* (1997) believe that complex water resources planning problems rely heavily on systems thinking, which they describe as a paradigm consisting of sets of objects forming systems and relationships between these systems, and that object-oriented programming is a suitable means of representing systems thinking. Wang *et al.* (2005a) note that in addition to the object-oriented approach enabling easier model extension, it also enables complex natural systems to be more closely replicated due to the linking of objects and their actions. Simonovic *et al.* (1997) state that the object-oriented modelling is a powerful approach for use in modelling for water resources planning and facilitates water resources decision making through the transparency achieved by separating policy from data. Bian (2007) believe that object-orientation has provided researchers with a new approach that enables them to model systems that would have previously been difficult to represent.

One aspect of the object-oriented approach that developers of water resource models need to consider carefully is the way in which objects are used to represent continuous space and time. Bian (2007) notes that in Geographic Information Systems (GISs) there is partitioning of continuous geographic space and that this is due to the discrete nature of computer representation, but questions whether these artificially partitioned spatial phenomena are always best represented as objects. Bian (2007) later concludes that the conceptual assumption in object-orientation that the real-world is composed of objects, is not always true in the case of spatial representation. Bian (2003) comments on the different approaches to representing space in models and states that the patch approach in which space is partitioned according to the form and function of landscape features is well suited for modelling spatially varied environments composed of landscape features where each is assumed to be homogeneous and stable. Galton (2004) discusses the problem posed by the fact that many real-world phenomena vary in both space and time, and questions how these multi-aspect phenomena are best represented. He starts by explaining that GIS typically uses a field approach, where attributes are associated with a spatial location, then from a time perspective, suggests a similar snapshot approach where attributes are associated with a point in time, and finishes by suggesting that the object approach, where attributes are associated with an object in which both time and space are attributes.

3.4 Applications in Water Resource Models

Wang *et al.* (2005b) state that in 2005 there were relatively few applications of object-orientation in hydrological models, and observe, together with Wang *et al.* (2005a), that although there were some applications of object-oriented programming in hydrological modelling, there was no detailed discussion of object-oriented design principles and how to implement these in hydrological models. A few examples from the literature, of models that have used the object-oriented approach, are briefly described in this section.

3.4.1 *ACRU* Model

The *ACRU* model is described in Schulze *et al.* (1995) as a physical conceptual agrohydrological model operating at a daily time step and is sensitive to climate, land cover/use and land management practices. Kiker *et al.* (2006) explain that the *ACRU* model was restructured using an object-oriented approach to provide more flexibility in the representation of spatial configurations and to provide a more flexible and extensible code structure. Details of the object-oriented structure of the restructured *ACRU* model can be found in Clark *et al.* (2001), Kiker and Clark (2001b) and Kiker *et al.* (2006). The object orientated version of the *ACRU* model is based on three main object classes: Components, Data and Processes. The Component classes are used to represent the physical entities of the hydrological system, both the spatial entities, such as catchments and dams, and, in some cases, sub-components of these spatial entities, such as climate vegetation and soil. All Component classes are based on the abstract *CComponent* class. Each Component object may contain a list of attributes represented by Data objects and a list of process algorithms represented by Process objects. Typically in object-orientation, attributes and behaviour are represented in classes as instance variables and methods respectively, but the *ACRU* model uses Data and Process objects mostly to facilitate model extensibility. All Data classes are based on the abstract *DData* class, and not only store attribute values but also include functionality such as range checking. A special abstract *DFluxRecord* class, is a subclass of *DData* and is used as the base for a set of Data classes representing the state of resources such as water and nutrients. All Process classes are based on the abstract *PProcess* class, and contain algorithms for simulating individual hydrological processes. At each daily time step a list of Processes objects is executed for each spatial Component object.

The extensibility of the object-oriented code structure has been demonstrated through the addition of a Nitrogen and Phosphorous module (Campbell *et al.*, 2001), natural vegetation,
herbivore and fire module (Kiker and Clark, 2001a), code to model shallow water-table flatwood areas in Florida (Martinez *et al.*, 2008), a river and dam operation module (Butler, 2001) and an irrigated sugarcane module (Moult, 2005). Martinez *et al.* (2008) found that the object-oriented design of the model to be flexible, modular and easily extensible enabling it to respond to evolving modelling concepts and needs. Campbell *et al.* (2001) state that one of the biggest advantages of the object-oriented design is the ease with which a process can be modified, removed or replaced with a different representation of the same process.

3.4.2 RHESSys Model

The Regional Hydro-Ecological Simulation System (RHESSys) is described by Band (2000) and Tague and Band (2004) as a spatially distributed hydro-ecological model for simulating water, carbon, and nutrient cycling and transport in catchments at a hillslope level. Spatial landscape entities are represented by an object-oriented spatial containment hierarchy consisting of Basins, which contain Hillslopes, which contain Climate Zones, which contain Patches, and where Patches may contain one or more non-spatial vertical Canopy Strata layers (Band *et al.*, 2000; Tague and Band, 2004). Processes are defined as methods within the specific landscape entity classes, corresponding to the scales at which they occur, resulting in a system of processes occurring at a range of spatial and temporal scales (Band *et al.*, 2000). Spatially distributed state variables are not within the spatial landscape classes, but are instead stored in linked, spatially defined lists in the *DataFactory* to facilitate the calculation of fluxes of water and dissolved components which are solved over flow fields using arrays of hydraulic potential values defining gradients scales (Band *et al.*, 2000).

3.4.3 OBJTOP Model

The OBJect-oriented TOPographic-based (OBJTOP) hydrological model is based on the concepts of TOPMODEL (Beven and Kirkby, 1979), which is described as a semi-distributed catchment scale hydrologic model in which topography is assumed to be the main driver of water flow through upland catchments, together with heterogeneous rainfall and soil type (Wang *et al.*, 2005a; Wang *et al.*, 2005b). Wang(2005a) explain that an object-oriented design approach was used for OBJTOP, the concept of inheritance was used to study individual processes at multiple levels, where processes are represented as objects, and the concept of aggregation was used to study process interactions. Wang (2005a) and Wang (2005b) explain that catchments are represented by a *Watershed* class which contains five subclasses *Precipitation*, *Vegetation*, *Evapotranspiration*, *Soil* and *Channel*. The *Precipitation* class contains component *Rainfall* and *Snowfall* classes and is also a generalisation of these two classes. The *Soil* class has subclasses *Upper Soil* and *Lower Soil*, where *Upper Soil* has subclasses *Surface* and *Root Zone*, and *Lower Soil* has subclasses *Unsaturated Zone* and *Saturated Zone*. Both the *Soil* and the *Channel* class can contain a *Topography* class. Wang (2005b) explain that all these classes act as super classes for a collection of classes representing specific processes. Wang (2005a) concludes that the object-oriented OBJTOP provides a catchment hydrological model that includes a flexible set of assumptions regarding the principal processes facilitating its use in catchments with different conditions, and that this object-oriented design has enabled complexity to be constrained by increasing simulation flexibility, without increasing the complexity of process representation.

3.4.4 RSM Model

The Regional Simulation Model (RSM) is described by Lal *et al.* (2005) as an object-oriented, physically based, hydrologic model designed for the South Florida region. Lal *et al.* (2005) explain that RSM uses a finite volume method to simulate two-dimensional surface and groundwater flow using unstructured triangular or rectangular mesh discretisations. According to Lal *et al.* (2005), the object-oriented design of RSM enables the hydrologic system to be decomposed based on a few simple abstractions enabling a single numerical scheme to be used for the equations describing all flow types. The *Water Body* class is the superclass for the *Cell* class representing discretised cell elements, the *Segment* class representing canal segments, and the *Lake* class representing lakes which store water. The *Water Body* class represents the mechanisms for moving water between water bodies, and has subclasses *Pipe Flow*, *Weir Flow* and *Single Control*. A set of stage-volume-relationship functions map between stages and volumes in water bodies. The *Pseudo Cells* class, with subclasses *Overland Flow*, *Urban Area* and *Agricultural Area*, represents the local hydrologic function in the water bodies and computes their recharge. Lal *et al.* (2005) conclude that the object-oriented design of RSM resulted in a robust and extensible software architecture.

3.4.5 RiverWare Model

RiverWare is a generalised, flexible modelling tool for simulation and optimisation of river and reservoir systems (Zagona *et al.*, 2001; Frevert *et al.*, 2006; Valerio *et al.*, 2010). RiverWare is based on an object-oriented framework (Zagona *et al.*, 2001) and models systems using networks of connected objects representing natural and artificial features such as reservoirs, river reaches, canals, groundwater and water users (Frevert *et al.*, 2006; Valerio *et al.*, 2010). Each object has an identity and contains attribute data and algorithms representing physical processes (Frevert *et al.*, 2006). Information is passed from one object to another through links between specific data attributes in the related objects (Frevert *et al.*, 2006).

4. MODEL INTEGRATION

IWRM requires integrated water resource assessment of complex hydrological systems, and it is unlikely that a single model will be able to adequately represent all components of a water resource system, which may include different scientific disciplines, different spatial and temporal scales, varying data availability and a variety of modelling objectives and stakeholders (Blind and Gregersen, 2005; Moore and Tindall, 2005; Gregersen *et al.*, 2007; Castronova and Goodall, 2010). The solution to real-world problems through modelling most often requires integrated analyses, which in turn requires linking of models (Kokkonen *et al.*, 2003). Kralisch *et al.* (2005) state that new challenges in sustainable water resource management require integrated, flexible hydrological models to simulate both quantitative and qualitative aspects of the hydrological cycle with a suitable degree of certainty. Existing models are often developed for, or have strengths, in specific domains within the hydrological system and integration of models is a popular solution in attempting to model complexity (Moore and Tindall, 2005; Barthel *et al.*, 2006). Though Kralisch *et al.* (2005) and Hoheisel (2002) point out that many existing models were developed for specific scales and purposes and are often coded in different programming languages or run on different operating systems, and cannot be easily adapted for integration with other models. Krause *et al.* (2005) state that combining and implementing approaches from natural and social sciences is a challenge to be faced in developing models and their application for IWRM.

4.1 Challenges in Model Integration

When coupling models, one of the main challenges is to understand and define the dependencies between models (Hoheisel, 2002). There are also both technical and conceptual constraints to be overcome when attempting to integrate models from different disciplines. Technically, existing environmental models are not generally designed to communicate with other models within the same discipline, let alone communicate with models from other disciplines, and conceptually, different models are often based on different ontologies due to different disciplines having different views of the natural environment, and differences in the way concepts are expressed in computer code (Argent, 2004). Different scientific disciplines approach system complexity and diverse scales in various ways, and use different modelling techniques and approaches to model design (Krause *et al.*, 2005). Integrated models must be compatible in terms of spatial and temporal scales and strategies for validation of both individual models and the integrated collection of models are necessary

(Barthel *et al.*, 2006). Integration of models and modelling approaches for IWRM has led to the development of integrated modelling frameworks, model interfacing specifications and modular modelling systems (Krause *et al.*, 2005). Gregersen *et al.* (2007) noted that some existing hydrological decision support systems were based on fixed combinations of specific hydrological and hydraulic models, but that limited supported combinations sometimes resulted in compromises being made in representing the hydrological system being modelled. Krause *et al.* (2005) noted that there were two main research and development paths being followed with regard to model integration: (i) direct integration of whole models through implementation of a model interface specification, and (ii) modular modelling systems where modules representing individual processes are combined to create custom models, where both approaches have advantages and disadvantages.

4.2 Simple Model Integration

As expressed by Krause *et al.* (2005), one of the simplest ways to combine models and modelling approaches from different domains or disciplines is the coupling of whole standalone models. There are many methods by which models can be coupled and these differ in their degree of complexity and the degree of interaction and feedback that can take place between the coupled models (Krause *et al.*, 2005). At the most basic level, model coupling involves using the output from one model as input for another model. This approach could be termed as running models in series, that is, each model is run independently for the full time period with the simulated output from one model used as input to the other model. The advantages of this approach are that it is simple and does not require any changes to the models used. There are two main problems with this approach, firstly the effort required to reformat the output from one model to be suitable for use as input for the other model, and secondly as stated by Krause *et al.* (2005), potential interactions and influences between the systems represented by the coupled models can only be realised in one direction, meaning that feedbacks cannot be modelled. The first problem can be overcome if both models use the same data input and output format.

4.3 Model Integration Using Modelling Frameworks

A recent trend has been the development of integrated modelling frameworks or decision support systems such as LIANA (Hofman, 2005), SPATSIM-HDSF (Clark *et al.*, 2009) and DeltaShell (Donchyts and Jagers, 2010). Integrated modelling frameworks typically include common data storage and formats, common data editing tools, and common spatial and temporal data visualisation and analysis tools. These modelling frameworks provide a modelling environment within which model users can operate without having to learn new user interfaces, data editing and analysis tools for each model, and enable model developers to concentrate on the science behind their models instead of having to reinvent the common functionality that is part of these modelling frameworks. Individual models would need to be modified to read from and write to the framework's data format, and would then benefit from being able to use the common tools within the framework. Integrated modelling frameworks assist in standardising the way in which models are run and resolve the problem of having to translate the output data format from one model to the input data format of the receiving model, but in general, models would still have to be run in series and therefore the problem of not being able to model feedbacks between the models would still exist. However, in some cases, for example the DeltaShell environment, these environments may include some means of directly coupling models (Krause *et al.*, 2005). Gärtner *et al.* (2001) state that there is a need for a means of enabling integration of data without requiring a central database.

4.4 Coupling Models

Running one or more models in parallel requires each model to be run one time step at a time, with values of variables being exchanged between models at each time step. One method of enabling two models to run in parallel would be to modify two or more specific models to communicate with each other either directly or via a common data repository on a time step-by-time step basis. When coupling two or more models in this manner, the computation order and protocols for data format and transfer have to be considered (Krause *et al.*, 2005). In order for this to work each model must have some means of being instructed to run each individual time step and there needs to be some sort of controller that commands each model, or a portion of each model, to run for the required time step. Alternatively, for the models to communicate directly with each other, they each need to provide some sort of publicly accessible interface, for example the Component Object Model (COM) interface standard, and the interface selected needs to be compatible with the operating platform and programming language of all of the models to be linked. The .Net programming platform has, in some respects, replaced COM by enabling compatibility between software modules written in different .Net programming languages. This linking approach requires the models to be modified to implement the interface standard, which may not be possible if the models are proprietary. This approach has the advantage that feedbacks can potentially be modelled, and though the models will need to be modified, legacy models can be linked without being completely re-written and thus retain their identity and in-built integrity. A disadvantage of this approach is that, though the specific models have been linked, further modifications may be required if another model needs to be linked into the suite of models.

4.5 Model Interface Specifications

As noted by Krause *et al.* (2005), one of the recent model integration development paths for integrated modelling has been the development of model interface specifications such as OpenMI (Blind and Gregersen, 2005; Gregersen *et al.*, 2005; Moore and Tindall, 2005; Gregersen *et al.*, 2007; Knapen *et al.*, 2009), Common Component Architecture (CCA) (Bramley *et al.*, 2000; Armstrong *et al.*, 2006) and the High Level Architecture (HLA) (Lindenschmidt *et al.*, 2005). A model interface standard consists of a set of software interfaces that must be implemented by the model that is to be made compliant with the standard. This concept of some sort of interface standard which must be adhered to is in some ways similar to the modularisation approach, except that it does not require the modularisation of legacy models. Implementation of the interface standard can be achieved in two ways, either by implementing the interface directly in the model code or creating a wrapper around the model. In the latter, the wrapper is compliant with the standard and has internal links to the wrapped model; however, the model may still need to be modified to some extent. Each model must declare sets of publically visible input and output variables. Feedbacks may be modelled if the model interface standard permits this. The models would be setup individually through their respective user interfaces. Links would then be created between appropriate variables in each model. It is important to note that it is specific applications of each model that are linked, not the model engines themselves.

Krause *et al.* (2005) conclude that though coupling models by means of model interface standards requires some effort to adapt the models, the advantages are increased flexibility, the ability to model more complex interactions and the ability to perform more detailed analyses of the coupled models. Other advantages of this approach are that the identity and integrity of legacy models are maintained, and their marketability is improved through their ability to link to other models using the same interface specification. Krause *et al.* (2005) concluded that, at that time, the OpenMI approach to model coupling was the most sophisticated. Hoheisel (2002) state that tight coupling of models usually uses shared memory to communicate between models usually coded in the same programming language and requiring a lot of effort to achieve integration, while loose coupling is more flexible, often using asynchronous communication between models.

4.6 Modular Modelling Frameworks

The other main development path for integrated modelling noted by Krause *et al.* (2005) has been the modularisation of models and the development of modular modelling frameworks such as such as MMS (Leavesley *et al.*, 2002), OMS (Ahuja *et al.*, 2005; Kralisch *et al.*, 2005), TIME (Rahman *et al.*, 2003; Argent and Rizzoli, 2004; Rahman *et al.*, 2004; Rahman *et al.*, 2005; Murray *et al.*, 2007) and LIQUID (Branger *et al.*, 2010a; Branger *et al.*, 2010b). Water resources models are typically structured into software components of some description that represent one or more hydrological processes. Thus, the concept of modularising legacy models into collections of modules representing individual hydrological processes makes a certain amount of sense. The modular modelling frameworks typically specify some sort of interface which each module must adhere to. Each module must declare sets of publically visible input and output variables. Several modules can then be linked within the appropriate modelling framework to create a custom-built model. Some sort of controller is usually required to setup the model and to coordinate calls to the various modules. The advantage of the modularisation approach is that custom-built models can be created to meet the requirements of specific modelling projects. The disadvantages of this approach are that there is a difference in the skills required by a model builder and a model user, and that the developers of legacy models must adopt one modular modelling framework. Feedbacks can be modelled if the controller and the module interface permit this, although the modularisation in itself may be sufficient for feedbacks to be modelled. Jones *et al.* (2001) conclude that a modular modelling approach, where new model components can be easily added, maintained and modified, facilitates integration of knowledge from different disciplines.

5. DISCUSSION AND CONCLUSIONS

Water is a critical resource in South Africa and needs to be managed efficiently and in a holistic manner to ensure equitable and sustainable use of this resource. These requirements are clearly expressed in South Africa's National Water Act, which is recognised as requiring new and innovative approaches to managing the nation's water resources. The complex nature of natural hydrological systems combined with complexity introduced by anthropogenic developments, increasing demands on already strained water resources, greater emphasis on sustainable environmental management and the recognition of the need for IWRM requires innovative and integrated modelling tools to assist water resource managers in understanding and quantifying the water resources under their control.

South Africa requires water resource modelling tools that are suitable for both planning and operations, accounting for all aspects of water resource systems in a holistic manner. These modelling tools need to be process-oriented and capable of representing real-world complexity at appropriate spatial and temporal scales. In particular, modelling for operations, needs to take place at a fine time scale, such as daily, to better represent process complexity and account for operational daily water use requirements such as irrigation and environmental flows, and to account for the lag and attenuation of flows through the flow network. Modelling at finer spatial scales is required to better represent spatial heterogeneity and complexity within catchments and to facilitate the allocation of water use licences at an individual water user level.

Two compatible means of addressing complexity in water resources modelling are recommended, object-orientation and integration of models from different water resource domains. Object-oriented design and programming enables model concepts and code to be structured in a more natural and intuitive manner, and results in models that are more flexible, extensible and modular. IWRM will require integration of models developed for specific domains, for use by multi-disciplinary modelling teams to combine hydrology with, ecological, economic, political and social aspects of water resources management. Ideally these models need to be integrated in such a way that feedbacks between the interrelated domains can be represented. The models adopted for use in South Africa should be suitable for local conditions, have access to local data sets and should be supported in South Africa by the developers. Ideally these models should all implement a common model interface standard, enabling them to be coupled to support IWRM.

6. PROJECT PROPOSAL

The problem presented in Chapter 1 can be summarised as the need for innovative and integrated modelling tools to assist water resource managers in South Africa in understanding and managing the complex water resource systems under their control in a holistic manner. This project proposal is intended to outline how part of this problem can be addressed through the use of object-oriented design and programming in the development of modelling tools to represent the real-world complexity water resource systems, and the use of a model interface standard to enable integration of legacy models from different disciplines.

6.1 Research Question

Can a process-oriented, hydrological model suitable for both water resource planning and operations modelling, be developed for application in South Africa to represent real-world complexity, through the use of the object-oriented design and programming approach, and the implementation of a model interface standard enabling it to be coupled with models representing other domains within a water resources system, to support integrated water resources management?

6.2 Methodology

The study will start with the identification of the water resource modelling requirements to improve water resource modelling as a tool for water resource managers in South Africa. These water resource modelling requirements are expected to include:

- Modelling at a finer time scale, such as daily, to better represent process complexity and account for operational daily water use requirements such as irrigation and environmental flows, and to account for the lag and attenuation of flows.
- Modelling at finer spatial scales to better represent spatial heterogeneity and complexity within catchments and to facilitate the allocation of water use licences at an individual water user level.
- Modelling dynamic changes within catchments.
- Modelling water quality as well as water quantity.
- Compatible modelling tools and methodologies for both planning and operations.
- Adoption of new modelling technologies such as object-orientation.
- Integration of water resources models that include ecological, economic and social aspects.

- Use of remotely sensed and forecast data for modelling.

Based on the water resource modelling requirements identified, the two primary objectives of the study are to revise and further develop the design of object-oriented structure of the *ACRU* agrohydrological model, and to implement a model interface standard enabling it to be coupled with other water resource related models. The *ACRU* model developed within the School of BEEH has been selected for the study as it is a daily, process-based, object-oriented hydrological model that has been developed in South Africa and is suitable for modelling South African hydrological conditions. It is also of strategic interest to the School and possibly South Africa, for the *ACRU* model to be updated and further developed for use in the assessment and management of the country's water resources.

The purpose of the *ACRU* model and its development history will be briefly documented. The object-oriented design of the *ACRU*2000 version of the *ACRU* model will be described and then evaluated regarding its suitability to meet the identified modelling requirements. Based on this evaluation recommendations for improvements to the object-oriented design of the model will be proposed. It is expected that the two key areas of concern will be the structure of the model input data and the suitability of the model for use in an operational modelling context. The *ACRU*2000 version of the *ACRU* model uses simple text based model input files which do not enable the object-oriented structure of the model to be used to its full potential, and it has long been recognised that the model input files need to be structured using object-oriented principles to be compatible with the model. The *ACRU* model has traditionally been used for planning purposes and is expected to require some modification and further development to make it suitable for operational modelling, including hot-starting, persistence of state variables and better handling of time-series data to accommodate the use of remotely sensed and forecasted data. The requirement for the *ACRU* model to be integrated with other models representing different domains will also be discussed.

An object-oriented data input structure for the *ACRU* model will be designed and implemented enabling the full potential of the object-oriented structure of the *ACRU* model itself to be leveraged. The new model input file structure should address four main requirements:

- Provide an object-oriented input file structure to complement the object-oriented structure of the *ACRU* model thereby enabling the restructured model to be used to its full potential.

- Provide a data model that is extensible such that new model parameters or variables can be accommodated without changes to the data model or to the software utilities that read from or write to the data model.
- Provide a structure for storing actual data values or references to where data values are stored, such as in a text file or database.
- Provide a structure for storing additional information that describes the model parameters or variables for use in graphical user interface software and other modelling tools developed to support the *ACRU* model.

It is proposed that the new model input file structure for the *ACRU* model be implemented using the Extensible Modelling Language (XML) as XML files are essentially text files and therefore platform independent, yet enable data to be structured in a manner that reflects the structure of the model, are extensible and can be easily serialised into memory to populate the model. Another advantage of XML files is that their structure may be declared in, and checked against, an XML Schema file which acts as a form of XML file template. It is likely that two separate XML files and associated XML Schemas will need to be designed. One file to store actual model input data or references to data, which will change for each model setup, and a second file to store additional information describing the model parameters or variables, which will be the same for each model setup. The new model input structure is expected to enable use of remotely sensed and forecast data for modelling, enable relevant model parameters and variables to be constant, dynamic or time series, and support the specification of scenario data.

Refinement and further development of the object-oriented design of the *ACRU* model will take place followed by implementation in the model code. The following refinement and further development is envisaged:

- Simplification of the Component class hierarchy and removal of redundant Component classes
- Complete restructuring of the Data class hierarchy to improve the way in which data, especially time series data is managed within the model, including the persistence of state data, the ability to handle breakpoint time series and improved functionality to deal with time-series of different types and intervals.
- Currently resources such as water and nutrients are modelled as special types of Data class known as flux records. Conceptually these resources are more than just state data but can also not be represented by Component objects which are conceptually discrete. It is proposed that a

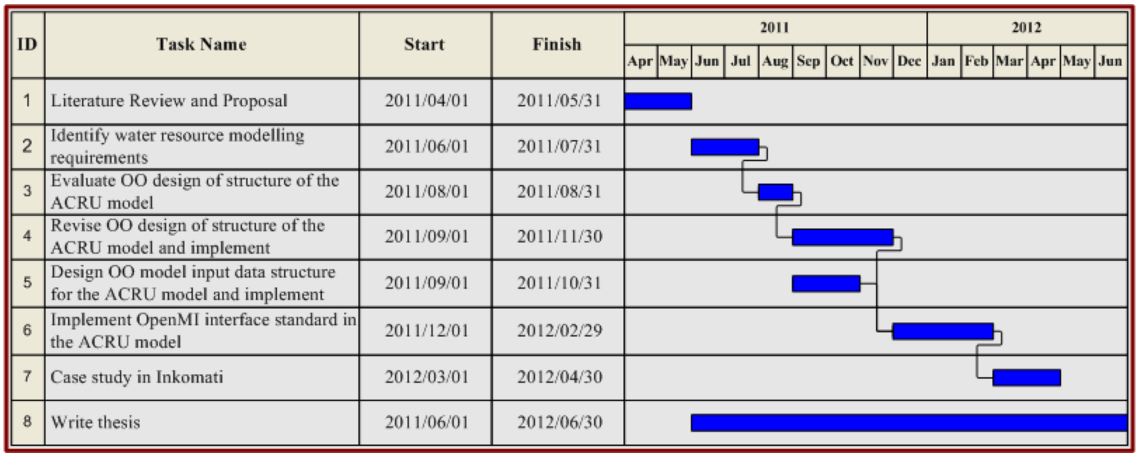
Resource class hierarchy be introduced to the *ACRU* model to improve the way in which these resources are conceptualised and managed within the model.

- The representation of hydrological processes as objects in the *ACRU* model makes the model easily extensible. Currently each spatial Component contains an ordered list of Process objects which are executed for each time step during a simulation. This arrangement works well in most cases, except where there are feedbacks between different spatial Component objects. A different approach to the way in which Process objects are executed is required and will be investigated.
- For operational modelling and when running the *ACRU* model coupled in parallel with another model it will be necessary to be able to hot-start the model and run it for one time step at a time. This will require some changes to the model which is currently designed to run the full simulation period and has no means of instructing the model to run one time step at a time. In addition several Process classes contain instance variables representing temporary process related state variables, and so these instance variables will need to be removed and replaced by Data objects.
- Some variables in hydrological modelling, for example landuse variables, are generally assumed to remain constant during a simulation, however, catchments develop over time and for long simulation periods this assumption may not be valid. The *ACRU* model previous included a means of setting up certain model variables to change dynamically during a simulation. This functionality needs to be included in the object-oriented version of the *ACRU* model, but improved to make it easier to use and so that variables can be specified and used in the model in the same manner, no matter whether they are constant or dynamic.
- The ability to be able to easily and efficiently setup and run a range of different scenarios would be useful when modelling for water resources planning and operations to answer "What if?" type questions. A means of setting up scenarios where a base scenario may have sub-scenarios containing only the differences is envisaged and would need to be implemented in both the *ACRU* model and the model input files.

It is conceptually and financially impractical to develop the *ACRU* model to include all the domains involved in water resource systems. However, there would be great value in implementing a suitable model coupling mechanism in the *ACRU* model through which it could be coupled to other existing models representing different domains. Clark *et al.* (2011) compiled a review and evaluation of methods of linking models, particularly methods suitable for use in the *ACRU* model. Clark *et al.*

(2011) stated that, of the systems reviewed, and in order of preference, the OpenMI, TIME and OMS systems were most suitable and should be evaluated further. It is recommended that the OpenMI system be implemented in the *ACRU* model as it a de facto standard as it has been widely adopted and implemented in other water resource related models and tools, is suitable for implementation in legacy models such as *ACRU* without requiring the model to be modularised, has been well documented and is supported by the OpenMI Association. The details of the OpenMI standard and supporting implementation classes will be investigated and a decision will be required as to whether the OpenMI standard will be implemented directly in the *ACRU* model or through a wrapper around the ACRU model. A technical issue that will need to be investigated is how an OpenMI compliant Java based *ACRU* model could be coupled to a model that is also OpenMI compliant but on the .Net platform.

The revised *ACRU* model, including the new model input data structure and model coupling mechanism will be demonstrated in a case study on a selected catchment in the Inkomati Water Management Area. A Gantt chart showing the expected timeline for the work proposed in this project is shown in Figure 6.1.



ID	Task Name	Start	Finish
1	Literature Review and Proposal	2011/04/01	2011/05/31
2	Identify water resource modelling requirements	2011/06/01	2011/07/31
3	Evaluate OO design of structure of the ACRU model	2011/08/01	2011/08/31
4	Revise OO design of structure of the ACRU model and implement	2011/09/01	2011/11/30
5	Design OO model input data structure for the ACRU model and implement	2011/09/01	2011/10/31
6	Implement OpenMI interface standard in the ACRU model	2011/12/01	2012/02/29
7	Case study in Inkomati	2012/03/01	2012/04/30
8	Write thesis	2011/06/01	2012/06/30

Figure 6.1 Gantt chart showing the expected timeline for the proposed work

6.3 Contribution to New Knowledge

The proposed project is expected to contribute to new knowledge in several areas. First, though the use of the object-oriented design approach in water resource models is not new, there are aspects of the proposed object-oriented design of the structure of the *ACRU* model that are different to that used in other object-oriented water resource models described in literature. The object-oriented models in literature appear to either use objects to represent the physical entities of the system being modelled with attributes and processes specified within these objects using instance variables and methods, or use objects to represent processes. The object-oriented structure of the *ACRU* model uses Component objects to represent the physical entities of the system being modelled as well as Data objects to represent the attributes and Process objects to represent the processes, to facilitate extensibility and flexibility of the model. In addition, it is proposed that resources such as water, nutrients, biomass and sediment will be represented by Resource objects, which is an approach that has not been found in literature. Further, the proposed design for an object-oriented model input data structure for the *ACRU* model is expected to be highly innovative and no similar data model has been found in literature. The *ACRU* model does not currently include a mechanism for coupling to other models and therefore implementation of OpenMI in *ACRU* can be regarded as new work. The development of a daily, process based, object-oriented, hydrological model developed for use in South Africa and which is suitable for both water planning and operations modelling and is capable of being integrated with other compatible models is expected to be unique in South Africa.

6.4 Resources and Equipment

The proposed work will be entirely computer based and will therefore require the use of a computer and suitable software. The majority of the proposed work will overlap with the work to be completed within WRC project K5/1951 and the candidate will require sufficient time away from other duties within the School of BEEH to complete the work required by the contract for the WRC project. Some advice may be required from experts in computer science.

6.5 Other Considerations

The PhD candidate is not aware of any health, safety, environmental or ethical issues that need to be considered. The IP rights of the *ACRU* model belongs jointly to the WRC who funded a large part of its development, the School of BEEH at UKZN who was responsible for a large part of the research and development, and the individual researchers who have developed process algorithms and model infrastructure.

7. REFERENCES

Ahuja, LR, Ascough Ii, JC and David, O. 2005. Developing natural resource models using the object modeling system: feasibility and challenges. *Advances in Geosciences* 4: 29-36.

Alfredsen, K and Saether, B. 2000. An object-oriented application framework for building water resource information and planning tools applied to the design of a flood analysis system. *Environmental Modelling & Software* 15 (3): 215-224.

Argent, R, Vertessy, R and Watson, F. 2000. A framework for catchment prediction modelling. *Proceedings of 26th National and 3rd International Hydrology and Water Resources Symposium*, Perth, Australia, 706-711. The Institution of Engineers.

Argent, RM. 2004. An overview of model integration for environmental application - components, frameworks and semantics. *Environmental Modelling & Software* 19 (3): 219-234.

Argent, RM and Houghton, B. 2001. Land and water resources model integration: software engineering and beyond. *Advances in Environmental Research* 5 (4): 351-359.

Argent, RM, Maul, C and Krämerkämper, T. 2002. Data frameworks for environmental modelling. In: eds. Rizzoli, AE and Jakeman, AJ, *Proceedings of the 1st Biennial Meeting of the International Environmental Modelling and Software Society (iEMSs)*, Lugano, Switzerland, 324-329.

Argent, RM and Rizzoli, AE. 2004. Development of multi-framework model components. In: eds. Pahl-Wostl, C, Schmidt, S, Rizzoli, AE and Jakeman, AJ, *Proceedings of the 2nd Biennial Meeting of the International Environmental Modelling and Software Society (iEMSs)*, Osnabrück, Germany, 365-370.

Armstrong, R, Kumfert, G, McInnes, LC, Parker, S, Allan, B, Sottile, M, Epperly, T and Dahlgren, T. 2006. The CCA component model for high-performance scientific computing. *Concurrency and Computation: Practice and Experience* 18 (2): 215-229.

Band, LE, Tague, CL, Brun, SE, Tenenbaum, DE and Fernandes, RA. 2000. Modelling watersheds as spatial object hierarchies: structure and dynamics. *Transactions in GIS* 4 (3): 181-196.

Barthel, R, Götzinger, J, Hartmann, G, Jagelke, J, Rojanschi, V and Wolf, J. 2006. Integration of hydrological models on different spatial and temporal scales. *Advances in Geosciences* 9: 1.

Beven, K and Kirkby, M. 1979. A physically based, variable contributing area model of basin hydrology. *Hydrological Sciences Journal* 24 (1): 43-69.

Bian, L. 2003. The representation of the environment in the context of individual-based modeling. *Ecological Modelling* 159 (2-3): 279-296.

Bian, L. 2007. Object-oriented representation of environmental phenomena: Is everything best represented as an object? *Annals of the Association of American Geographers* 97 (2): 267 - 281.

Blind, M and Gregersen, J. 2005. Towards an Open Modelling Interface (OpenMI) the HarmonIT project. *Advances in Geosciences* 4: 69-74.

Booch, G. 1994. *Object-oriented analysis and design with applications*. Second Edition. Addison-Wesley, Boston, USA.

Bramley, R, Chiu, K, Diwan, S, Gannon, D, Govindaraju, M, Mukhi, N, Temko, B and Yechuri, M. 2000. A component based services architecture for building distributed applications. *Proceedings of the 9th IEEE International Symposium on High Performance Distributed Computing*, Pittsburgh, Pennsylvania, USA, IEEE Computer Society.

Branger, F, Braud, I, Debionne, S, Viallet, P, Dehotin, J, Henine, H, Nedelec, Y and Anquetin, S. 2010a. Towards multi-scale integrated hydrological models using the LIQUID (R) framework. Overview of the concepts and first application examples. *Environmental Modelling & Software* 25 (12): 1672-1681.

Branger, F, Debionne, S, Viallet, P, Braud, I, Jankowfsky, S, Vannier, O, Rodriguez, F and Anquetin, S. 2010b. Advances in integrated hydrological modelling with the LIQUID framework. In: eds. Swayne, D, Yang, W, Voinov, A, Rizzoli, A and Filatova, T, *Proceedings of the 5th Biennial Meeting of the International Environmental Modelling and Software Society (iEMSs)*, Ottawa, Canada,

Butler, AJE. 2001. *The development and evaluation of an operating rule framework for the ACRU agrohydrological modelling system*. Unpublished MSc Eng dissertation. School of Bioresources Engineering and Environmental Hydrology, University of KwaZulu-Natal, Pietermaritzburg, RSA.

Campbell, KL, Kiker, GA and Clark, DJ. 2001. Development and testing of a nitrogen and phosphorus process model for Southern African water quality issues. ASAE Paper No. 012085. ASAE, St. Joseph, Michigan, USA,

Castronova, AM and Goodall, JL. 2010. A generic approach for developing process-level hydrologic modeling components. *Environmental Modelling & Software* 25 (7): 819-825.

Clark, DJ, Kiker, GA and Schulze, RE. 2001. Object-oriented restructuring of the ACRU agrohydrological modelling system. *Proceedings of the 10th South African National Hydrology Symposium*, Pietermaritzburg, RSA, 26-28.

Clark, DJ, Lutchminarain, A and Smithers, JC. 2011. *Integrated modelling for water resource planning and operations*. Report to the Water Research Commission, WRC Project K5/1951. School of Bioresources Engineering and Environmental Hydrology, University of KwaZulu-Natal, Pietermaritzburg, RSA.

Clark, DJ, Smithers, JC, Hughes, DA, Meier, KB, Summerton, MJ and Butler, AJE. 2009. *Design and development of a hydrological decision support framework.* Water Research Commission, Pretoria, RSA.

Colvin, J, Ballim, F, Chimbuya, S, Everard, M, Goss, J, Klarenberg, G, Ndlovu, S, Ncala, D and Weston, D. 2008. Building capacity for co-operative governance as a basis for integrated water resource managing in the Inkomati and Mvoti catchments, South Africa. *WaterSA* 34 (6): 681-689.

Cook, S and Daniels, J. 1994. *Designing object systems*. Prentice-Hall, UK.

Donchyts, G and Jagers, B. 2010. DeltaShell - an open modelling environment. In: eds. Swayne, D, Yang, W, Voinov, A, Rizzoli, A and Filatova, T, *Proceedings of the 5th Biennial Meeting of the International Environmental Modelling and Software Society (iEMSs)*, Ottawa, Canada, 1050-1057.

DWAF, 2004. National Water Resource Strategy. Department of Water Affairs and Forestry, Pretoria, RSA.

Egenhofer, M and Frank, A. 1992. Object-oriented modeling for GIS. *Urisa Journal* 4 (2): 3-19.

Elshorbagy, A and Ormsbee, L. 2006. Object-oriented modeling approach to surface water quality management. *Environmental Modelling & Software* 21 (5): 689-698.

Frevert, D, Fulp, T, Zagona, E, Leavesley, G and Lins, H. 2006. Watershed and River Systems Management Program: Overview of Capabilities. *Journal of Irrigation and Drainage Engineering* 132 (2): 92-97.

Galton, A. 2004. Fields and objects in space, time, and space-time. *Spatial Cognition & Computation* 4 (1): 39-68.

Garrote, L and Becchi, I. 1997. Object-oriented software for distributed rainfall-runoff models. *Journal of Computing in Civil Engineering* 11 (3): 190-194.

Gärtner, H, Bergmann, A and Schmidt, J. 2001. Object-oriented modeling of data sources as a tool for the integration of heterogeneous geoscientific information. *Computers & Geosciences* 27 (8): 975-985.

Gregersen, JB, Gijsbers, PJA and Westen, SJP. 2007. OpenMI: Open modelling interface. *Journal of Hydroinformatics* 9 (3): 175-191.

Gregersen, JB, Gijsbers, PJA, Westen, SJP and Blind, M. 2005. OpenMI: the essential concepts and their implications for legacy software. *Advances in Geosciences* 4: 37-44.

Hofman, D. 2005. LIANA Model Integration System - architecture, user interface design and application in MOIRA DSS. *Advances in Geosciences* 4: 9-16.

Hoheisel, A. 2002. Model coupling and integration via XML in the M3 simulation. In: eds. Rizzoli, AE and Jakeman, AJ, *Proceedings of the 1st Biennial Meeting of the International Environmental Modelling and Software Society (iEMSs)*, Lugano, Switzerland, 611–616.

Horak, J, Orlik, A and Stromsky, J. 2008. Web services for distributed and interoperable hydro-information systems. *Hydrology and Earth System Sciences* 12 (2): 635-644.

Javadi, S, Kiapasha, MS and Mohammadi, K. 2009. Object oriented simulation; Its application in water reservoir management and operation. *Journal of Agricultural Science and Technology* 11 (3): 331-340.

Jones, JW, Keating, BA and Porter, CH. 2001. Approaches to modular model development. *Agricultural Systems* 70 (2-3): 421-443.

Kiker, GA and Clark, DJ. 2001a. Development and testing of a natural vegetation, herbivore, and fire model for southern African rangeland management. ASAE Paper No. 017025. ASAE, St. Joseph, Michigan, USA,

Kiker, GA and Clark, DJ. 2001b. The development of a Java-based, object-oriented modeling system for simulation of southern African hydrology. ASAE Paper No. 012030. ASAE, St. Joseph, Michigan, USA,

Kiker, GA, Clark, DJ, Martinez, CJ and Schulze, RE. 2006. A Java-based, object-oriented modeling system for Southern African hydrology. *Transactions of the ASABE* 49 (5): 1419-1433.

Knapen, M, Verweij, P, Wien, J and Hummel, S. 2009. OpenMI–The universal glue for integrated modelling? *18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*, Cairns, Australia,

Kokkonen, T, Jolma, A and Koivusalo, H. 2003. Interfacing environmental simulation models and databases using XML. *Environmental Modelling & Software* 18 (5): 463-471.

Kralisch, S, Krause, P and David, O. 2005. Using the object modeling system for hydrological model development and application. *Advances in Geosciences* 4 (1-2): 75–81.

Krause, P, Kralisch, S, Flügel, W, Haas, A, Jaeger, C, Hofman, D, Pullar, D, Lotze-Campen, H, Lucht, W and Müller, C. 2005. Model integration and development of modular modelling systems. *Advances in Geosciences* 4: 1-2.

Kumar, M, Bhatt, G and Duffy, CJ. 2010. An object-oriented shared data model for GIS and distributed hydrologic models. *International Journal of Geographical Information Science* 24 (7): 1061-1079.

Lafore, R. 2002. *Object-oriented programming in C++*. 4th Edition. Sams Publishing, Indianapolis, Indiana, USA,

Lal, AMW, Van Zee, R and Belnap, M. 2005. Case study: Model to simulate regional flow in South Florida. *Journal of Hydraulic Engineering* 131 (4): 247-258.

Leavesley, G, Markstrom, S, Restrepo, P and Viger, R. 2002. A modular approach to addressing model design, scale, and parameter estimation issues in distributed hydrological modelling. *Hydrological Processes* 16 (2): 173-187.

Leone, A and Chen, DY. 2007. Implementation of an object oriented data model in an information system for water catchment management: Java JDO and Db4o Object Database. *Environmental Modelling & Software* 22 (12): 1805-1810.

Lindenschmidt, KE, Hesser, FB and Rode, M. 2005. Integrating water quality models in the High Level Architecture (HLA) environment. *Advances in Geosciences* 4: 51-56.

Liu, DF and Stewart, TJ. 2004. Object-oriented decision support system modelling for multicriteria decision making in natural resource management. *Computers & Operations Research* 31 (7): 985-999.

Manqoyi, N and Nyabeze, WR. 2006. *Situational assessment of operating rules for existing DWAF dams*. Makgaleng Projects CC, Midrand, RSA.

Martinez, CJ, Campbell, KL, Annable, MD and Kiker, GA. 2008. An object-oriented hydrologic model for humid, shallow water-table environments. *Journal of Hydrology* 351 (3-4): 368-381.

McCartney, M, Awulachew, S, Seleshi, Y, Prasad, K, King, J and Tarekegn, D. 2005. Decision support systems for dam planning and operation in Africa. *Challenge Program on Water and Food: Project Leaders Meeting*, Entebbe, Uganda,

McDonnell, J, Sivapalan, M, Vaché, K, Dunn, S, Grant, G, Haggerty, R, Hinz, C, Hooper, R, Kirchner, J and Roderick, M. 2007. Moving beyond heterogeneity and process complexity: A new vision for watershed hydrology. *Water Resources Research* 43 (7): W07301.

Mckenzie, RS and Van Rooyen, PG. 2003. Management of large water resources systems. *Water Supply* 3 (3): 297–304.

Molina, JL, Bromley, J, García-Aróstegui, JL, Sullivan, C and Benavente, J. 2010. Integrated water resources management of overexploited hydrogeological systems using Object-Oriented Bayesian Networks. *Environmental Modelling & Software* 25 (4): 383-397.

Moore, RV and Tindall, CI. 2005. An overview of the open modelling interface and environment (the OpenMI). *Environmental Science & Policy* 8 (3): 279-286.

Moult, NG. 2005. *The development of a catchment scale irrigation systems model for sugarcane*. Unpublished MSc Eng dissertation. School of Bioresources Engineering and Environmental Hydrology, University of KwaZulu-Natal, Pietermaritzburg, RSA.

Murray, N, Perraud, J-M, Rahman, J, Bridgart, R, Davis, G, Watson, F and Hotham, H. 2007. *TIME Workshop Notes 4.2*. CSIRO Land and Water, and eWater CRC, Australia.

NWA. 1998. *National Water Act, Act No. 36 of 1998*. Government Printers, Pretoria, RSA,

Pollard, S and du Toit, D. 2008. Integrated water resource management in complex systems: How the catchment management strategies seek to achieve sustainability and equity in water resources in South Africa. *WaterSA* 34 (6): 671-679.

Rahman, J, Perraud, J, Hotham, H, Murray, N, Leighton, B, Freebairn, A, Davis, G and Bridgart, R. 2005. Evolution of TIME. In: eds. Zerger, A and Argent, RM, *MODSIM 2005 International Congress on Modelling and Simulation*, Melbourne, Australia, 697-703. Modelling and Simulation Society of Australia and New Zealand.

Rahman, J, Seaton, S, Perraud, J, Hotham, H, Verrelli, D and Coleman, J. 2003. It's TIME for a new environmental modelling framework. In: ed. Post, DA, *MODSIM 2003 International Congress on Modelling and Simulation*, Townsville, Australia, 1727-1732. Modelling and Simulation Society of Australia and New Zealand.

Rahman, JM, Seaton, SP and Cuddy, SM. 2004. Making frameworks more useable: using model introspection and metadata to develop model processing tools. *Environmental Modelling & Software* 19 (3): 275-284.

Reitsma, R and Carron, J. 1997. Object-oriented simulation and evaluation of river basin operations. *Journal of Geographic Information and Decision Analysis* 1 (1): 10-24.

Rumbaugh, J, Blaha, M, Premerlani, W, Eddy, F and Lorensen, W. 1991. *Object-oriented modeling and design*. Prentice Hall, Englewood Cliffs, New Jersey, USA,

Schulze, RE, Angus, GR, Lynch, SD and Smithers, JC. 1995. ACRU: Concepts and structure. In: *Hydrology and agrohydrology: A text to accompany the ACRU 3.00 agrohydrological modelling system*. WRC Report No. TT69/95. Water Research Commission, Pretoria, RSA.

Shane, RM, Zagona, EA, McIntosh, D, Fulp, TJ and Goranflo, HM. 1996. Project object. *Civil Engineering* 66 (1): 61-63.

Shaw, EM. 1994. *Hydrology in practice*. Third Edition. Chapman and Hall, London, UK.

Silvert, W. 1993. Object-oriented ecosystem modelling. *Ecological Modelling* 68 (1-2): 91-118.

Silvert, W. 2001. Modelling as a discipline. *International Journal of General Systems* 30 (3): 261-282.

Simonovic, SP, Fahmy, H and El-Shorbagy, A. 1997. Use of object-oriented modeling for water resources planning in Egypt. *Water Resources Management* 11: 243-261.

Spanou, M and Chen, D. 2000. An object-oriented tool for the control of point-source pollution in river systems. *Environmental Modelling & Software* 15: 35-54.

Sydelko, PJ, Dolph, J, Taxon, T and Majerus, K. 1999. A dynamic object-oriented architecture approach to ecosystem modeling and simulation. *Proceedings of the 1999 American Society for Photogrammetry and Remote Sensing (ASPRS) Annual Conference*, Portland, Oregon, USA, 410-420.

Sydelko, PJ, Hlohowskyj, I, Majerus, K, Christiansen, J and Dolph, J. 2001. An object-oriented framework for dynamic ecosystem modeling: application for integrated risk assessment. *Science of the Total Environment* 274 (1-3): 271-281.

Tague, CL and Band, LE. 2004. RHESSys: Regional Hydro-Ecologic Simulation System-An object-oriented approach to spatially distributed modeling of carbon, water, and nutrient cycling. *Earth Interactions* 8: 1-42.

Valerio, A, Rajaram, H and Zagona, E. 2010. Incorporating groundwater-surface water interaction into river management models. *Ground Water* 48 (5): 661-673.

Wang, J, Endreny, TA and Hassett, JM. 2005a. A flexible modeling package for topographically based watershed hydrology. *Journal of Hydrology* 314 (1-4): 78-91.

Wang, J, Hassett, JM and Endreny, TA. 2005b. An object oriented approach to the description and simulation of watershed scale hydrologic processes. *Computers & Geosciences* 31 (4): 425-435.

Wegner, P. 1990. Concepts and paradigms of object-oriented programming. *ACM SIGPLAN OOPS Messenger* 1 (1): 7-87.

Wirth, N. 2006. Good ideas, through the looking glass. *IEEE Computer Society* 39 (1): 28-39.

Zagona, EA, Fuip, TJ, Shane, R, Magee, T and Goranflo, HM. 2001. RiverWare: A generalized tool for complex reservoir system modeling. *JAWRA Journal of the American Water Resources Association* 37 (4): 913-929.