

# Conceptual Microarchitectures for Hydrologic Simulation Models

**Urciuolo Adriana**

Universidad Nacional de la Patagonia San Juan Bosco  
Darwin esq. Canga- 9410 Ushuaia, Argentina, 54-2901-443533  
urciuolo@tdfuego.com

**Iturraspe Rodolfo**

Universidad Nacional de la Patagonia San Juan Bosco  
Darwin esq. Canga -9410 Ushuaia, Argentina, 54-2901-432217  
iturraspe@tdfuego.com

**Parsón Ariel**

Universidad Nacional de la Patagonia San Juan Bosco  
Darwin esq. Canga, 9410 Ushuaia, Argentina, 54-2901-430892  
a-parson@infovia.com.ar

## ABSTRACT

Mathematical Hydrologic models simulate real world environmental processes through different strategies. Each process is calculated by means of methods that utilize physical parameters for representing the real world system. Some parameters are obtained from tables, some of them are optimized and others may be calculated using environmental variables. Although the domain software provides a wide range of models, there is not a conceptual architecture that allows the maintenance of the vast knowledge about simulation strategies and parameters collected in environmental management organizations, facilitating the flexible simulation scenarios configuration. The present work shows how to face this problem by means of conceptual analysis models organized in the scope of a general architecture.

It's also possible for the given architecture, to analyze and define microarchitectures for software components related to particular problems.

In the present work, conceptual microarchitectures are defined to construct a knowledge level for hydrologic models systems starting from a general conceptual Environmental Information Systems architecture. To get the required flexibility for the conceptual and design models, high-level components are identified and different kinds of patterns are applied.

### **Keywords:**

Architecture, analysis patterns, simulation models, parameter.

## 1. INTRODUCTION

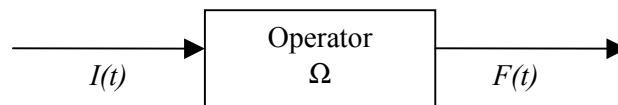
In the last years, great advances have been made in techniques for exploring the hydrologic environment, such as remote sensing, radar and others. Due to limitations of these techniques, there is only a limited range of measurements in space and time, which is normally stored in environmental organizations. Therefore, models of different types provide a means of quantitative extrapolation or prediction from the available measurements in space and time [6]. Models generally make assumptions about real world

processes in order to predict the behavior of systems under certain conditions to assess the likely impact of future hydrological change.

The ultimate aim of predicting using models is to improve decision-making about some hydrological problems in Water Resource Management, such as water resource planning, flood prediction, determination of risk areas, mitigation of contamination and so on.

Modeling has existed for at least one hundred and fifty (150) years. It is based on the idea of conservation of mass, momentum and energy. The governing equations for environmental models resulted from experimentation.

A hydrological model is a simplified representation of the real world system which aim is to study the system operation and predicting its outflow. Model inputs and outputs are hydrologic and climatic variables such as water flow, precipitation, temperature, evaporation and others. The model structure is a set of mathematical equations connecting inputs with outputs [11].



$$F(t) = \Omega I(t) \text{ system transformation equation}$$

As it was said, hydrological models serve a range of purposes but they are used primarily to estimate runoff from sequences of rainfall and the meteorological information needed to estimate potential evaporation. They can be used to estimate river flows at ungauged sites, fill gaps in broken records or extend flow records with respect to longer records of rainfall.

Generally, the relationship between rainfall and runoff is a central problem in hydrology. According to this criteria, models can be classified in empirical, conceptual or physically-based. Mathematical modeling of this relationship is often based on a black box type of model with less regard to the physics of the process. This is partly due to the complexity of the process and data availability. However there is a continuing interest among hydrologists to develop rainfall-runoff models with a theoretical structure based on physical laws. Powerful areally-distributed models are based on physical principles governing the movement of water within a catchment area, but they need detailed high-quality data to be used effectively. More commonly, simpler conceptual models are used to represent the basin as a whole. The main controls on water movement are represented by quasi-physical model elements whose action is governed by a set of model parameters. In some circumstances, these parameters can be adjusted to represent changes to land-use in the catchment area.

Computational models present a clear evolution process, due to the computational advances to which they were adapting. It's common, when tracing the development of models to distinguish five generations [1], but hydrologic simulation models as software products for decision-making arose in the fourth-generation, with some new requirements such as friendly job environments, use of models by non domain experts, etc., giving place to the creation of a new paradigm called hydroinformatics [1]

Despite its evolution, there are some critic questions to face in the software domain, related with the lack of a conceptual framework for Modeling in Water Resource Management. There are many different types of hydrologic models, considering the simulation strategies and methods used for flow computing, but they are often isolated from the Environmental systems, which provide basic services and information to them. The complex models used for hydrologic studies require extensive parameter input which decreases model accuracy. Sometimes it's a serious restriction for their use. On the other hand, modeling software usually is based over a static design model, which doesn't allow dynamically changing methods or parameters if they are not available or if they are not the appropriate for the real system simulation. In summary, there is not a domain software architecture that facilitates the maintenance of the hydrological

processes simulation methods and parameters knowledge for the flexible configuration of simulations scenarios and the integration of models to Environmental Information Systems.

High-level domain specific software analysis appears as a possible solution for such topics.

The problem may be characterized as follows:

- ✓ There is a great diversity of methods for each hydrological process simulation by means of different strategies (black box, conceptual and physically based models).
- ✓ There are different parameters for each type of hydrological structural component (lake, river, soil, etc.), each simulated process and selected method used by models. Some parameters can be obtained from tables, some of them can be calculated from environmental variables and others are obtained using optimization methods.
- ✓ Environmental management organizations collect important knowledge through the years about parameters that better fit the real systems conditions for a given region.
- ✓ Simulation methods and parameters may use the data from existing Environmental Information Systems for flow computing, but models are often acquired and used isolated from other systems.

The domain has been traditionally characterized by the study of efficient flow computing algorithms, but not of high-level solutions for the exposed questions. In the present work, it is proposed to introduce flexible software developing mechanisms, such as architecture, patterns and components in the construction of specific domain software.

Software architecture deals with the design and implementation of the high-level structure of the software [5]. Architectural descriptions are being recognized as an appropriate vehicle for codification and reuse of design knowledge. Software domain specific architectures provide an organizational structure for a family of applications, that allow to organize their development and sustain their evolution and reuse.

By other hand, for a given architecture style, it's possible to define a set of conceptual, design and architectural patterns, which act as "microarchitectures" [21]. They constitute appropriate techniques to solve particular problems, such as those exposed above.

Patterns are used in software engineering to allow the reuse of solutions that have been prove to be successful in the different software process stages. There are different types of patterns, which may be classified by many points of view, such as: abstraction level, scale ranges, dependency on a specific particular domain and software process stage [22].

This article provides conceptual microarchitectures for flexible hydrologic modeling software on the basis of a general conceptual architecture that facilitates simulation models integration to Environmental Information Systems. The remainder of this paper is organized as follows. Section 2 reviews software techniques for flexible software development. Section 3 presents the conceptual analysis process. Section 4 includes the proposed design models for flexible simulation scenarios configuration. Section 5 shows an example use case of the models. Section 6 is related with acknowledgements and section 7 presents conclusions and future works.

## **2. FLEXIBLE SOFTWARE: PATTERNS, ARCHITECTURE AND COMPONENTS.**

Architecture has emerged as a crucial part of the design process. Larry Bass [5] defines software architecture as: "The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them".

From the above definition, it is inferred that when proposing specific domain software architectures, some information is abstracted to concentrate in relations and high-level structures, providing enough information to constitute the basis for analysis.

It's possible to use different architectural views to enhance the understandability of the architecture and to focus on particular concerns separately. Some studies [20] propose the following three views: conceptual, logical and execution view.

The Conceptual Architecture identifies the high-level components of the system, and the relationships among them [4]. Its purpose is to direct attention at an appropriate decomposition of the system without delving into details. It consists of the Architecture Diagram (without interface detail) and an informal component specification for each one.

In Logical Architecture, the externally visible properties of the components are made precise and unambiguous through well-defined interfaces and component specifications, and key architectural mechanisms are detailed.

Execution Architecture is created for distributed or concurrent systems. The process view shows the mapping of components onto the processes of the physical system.

In the present work the emphasis is put on the conceptual architecture analysis. Defining this basic initial structure, then it's possible to develop microarchitectures for software components related with specific domain problems. This architecture may be refined in successive iterations of the software process.

As it was said, fundamental to conceptual software architecture is the structure of the system in terms of the primary structural elements or *components of the system*, and their interrelationships. The use of components allows identifying independent units [23] for new appropriate solutions, as well as the reuse of existing solutions.

Patterns have been used in different domains to define microarchitectures for particular problems. An *architectural pattern* [10] expresses a fundamental structural organization or schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them [3]. A *conceptual (or analysis) pattern* is a pattern whose form is described by means of terms and concepts from an application domain. Analysis patterns were proposed by Fowler [13] to make possible the reuse of conceptual modeling solutions. This term is utilized for patterns used to describe solutions for problems that appear during the requirement and conceptual modeling stage. These patterns facilitate the transformation from analysis to design model. A *design pattern* describes commonly recurring structure of communicating components that solves a general design problem within a particular context. A *design pattern* can be defined as a common solution to recurring problems in software design [15]. Provides a scheme for refining the subsystems or components of a software system, or the relationships between them.

Although there are few antecedents of patterns application in hydroinformatics systems [2] [24] [25], they are important for the study of specific environmental domain architectures, as it can be seen in recent researches related with Geographic Information Systems [19] [16].

### 3. CONCEPTUAL ANALYSIS

#### 3.1 Methodology

The methodology for the early stages of software development is based on the RUP (Rational Unified Process) Process Development [8], considering its basic features: case use driven, architecture centric, iterative and incremental.

First it is defined the general conceptual architecture to be used as a framework for hydrologic models software development. It is used an existing conceptual architecture for Hydrological Modeling integrated to Environmental Information Systems. It was developed on the basis of a Physical Water Resource Domain Model, which was used for the domain objects and analysis classes identification [7] [24].



























