Description of systems represented as processes networks using XML

H F Castro Silva¹, C A Parra Ortega² and R Reyes Pinilla²
¹ Grupo de Investigación Observatorio, Universidad Pedagógica y Tecnológica de Colombia, Sogamoso, Colombia
² Grupo de Investigación en Ciencias Computacionales, Universidad de Pamplona, Pamplona, Colombia

E-mail: hugofernando.castro@uptc.edu.co, carapa@unipamplona.edu.co

Abstract: In this article, a specification based on the extensible markup language (XML) is proposed and validated to describe systems whose representation is a stochastic networking process. Such concept is related to a kind of flow developed within networks process. Extensible Markup Language indicates how description of this type of systems would be. On the other hand, this definition of language by means of software is validated, and a utilization scenario is proposed to be applied in web-based simulation.

1. Introduction
Nowadays there exists a great quantity of applications to carry out tasks of modeling and simulation. Such applications allow to edit, to evaluate and to simulate models using diverse methodologies. On operations research field, to simulate systems represented by means of flow in networks exist some several proprietary applications. Nevertheless, it is difficult for a user of an application to execute or to compile the models developed in another application, as it implies the acquisition of a simulator. Likewise, these applications are designed for certain operating systems, implying problems of compatibility. The previous thing leads to the difficulty of exchanging models of this nature and does not take advantage of the Internet potential as for exchange of information and universal portability.

For this reason, it is intended to take advantage of methodology that facilitate the model standardization, especially using the extensible markup language (XML) with the double intention of improving the possibilities of portability among applications, and to promote the creation of web services whose purpose is simulating models codified with XML. Hereby, users would only be focused on the model, obviating the managing and learning curve of this application.

Based on the previous motivation with the experience obtained through the project execution on the operations research field, modeling and simulation and information management; a format of information exchange has been designed based on the extensible markup language (XML) to structure and store models which describe stochastic processes and discrete-events that could represent a flow of entities across networks. This format was proved with validators, as well as some examples, which is a step towards the design of web services able to simulate systems of this kind.
2. Review of the literature
The stochastic processes networks are a methodology for modeling systems based on the occurrence of discrete events as it is defined by several authors [1-4], which guide the evolution of the variables of system conditions.

2.1. Elements and relations in the models
A model, as an abstract representation of a real system, is made out of: 1) A set of definitions that allow to identify the elements that constitute the model, on this case would be the processes and associate resources; 2) A set of relations that specify the flow of entities across a network process; 3) The entities that circulate along the network, which use resources as they are moved from a process to other one. Besides representing a system, these models are simulated to obtain some measure of performance on entities, degree of utilization of the resources and queues in every process, among others.

2.2. Processes networks
Among the elements that constitute a dynamic system there is an established scheme in which the relations are represented by links joining them across lines of flow. This structure is given by the specification of the variables that appear in the dynamic system, which represent processes whose time of service behaves as a random variable.

Another fundamental component of this system regards some "entities", which are those objects that come to the system, and receive attention in the processes that are represented in the network of components. These entities circulate freely along the system and there are some examples: objects in a repairing workshop; people attending a bureaucratic system; pieces in a process of manufacture and ducts that are repaired by a team. In the Figure 1(a), a general scheme can be appreciated by the system represented under these conventions:

![Diagram](image.png)

**Figure 1.** (a) Scheme of a system represented by mean of process networks. (b) XML-based scheme to simulate the performance of these systems.

As shown in Figure 1(a), there are some entities arriving to the system, and they circulate across the processes following a directed flow, organized in a network. Some entities can wait in a process, while other entities are using the resources assigned to the processes, which are limited. Once some processes are completed, the entities leave from the system.

2.3. The extensible markup language and its role in the web-based simulation
With the rise of the Internet and its hypertext-based language (HTML), there arose the need to exchange structured information, such as records of transactions, invoices, and request orders, among others. The Web World Wide consortium established then the standard to create the Extensible Markup Language, known as XML [5], to satisfy these needs. Inside the characteristics of this language [5] are the following; [4,6]: It is extensible and allows to define new labels. If users need it, representation structured of the information; separation of the processing and presentation of the
information and the information XML is auto-descriptive, that is to say, it is intelligible for people, besides the machines.

Since the models in network process can be described by means of structured information, there is a potential use of XML in the codification of these models, and to facilitate its exchange among diverse applications as web-based simulators. Some specifications have been carried out successfully, even on simulation way. One under the approach of client-sided software developed for studies of tourism [7], another approach based on web services in the server-side is in use for education of intensive therapy in nursing [8]. A proposal of XML codification designed for networks was validated using Java [9]. The scheme showed in the Figure 1(b) indicates the wide use that can have a XML-based model in simulating the performance of networks process.

This scheme illustrates the different ways of using network models based on XML, so much for stand-alone use, as direct exchange among workstations, or in internet-based schemes, where a user provides a model to be executed by a web service and to visualize the results by means of a graphical interface. Given the technological current advances, it is expected that the mobile devices could execute some of these services.

3. XML specifications to describe models of processes network

The XML language needs a previous definition of the information that will be presented. This information is stored in the XSD (XML schema definition) files developed by the WWW Consortium. This specification defines the structure of the document and allows its validation [5,6,10,11].

3.1. Description of the stochastic networks process by means of XML

Modifying the initial proposal of Mercado [6], in order to offering more information about the behavior of service time, the scheme that describes the model as a process net is shown in Figure 2.

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://xml.netbeans.org/schema/red_schema"
  xmlns:tns="http://xml.netbeans.org/schema/red_schema"
  elementFormDefault="qualified">
  <xs:simpleType name="TipoEntero">
    <xs:restriction base="xs:integer"/>
  </xs:simpleType>
  <xs:simpleType name="TipoFlotante">
    <xs:restriction base="xs:float"/>
  </xs:simpleType>
  <xs:simpleType name="TipoNuevo">
    <xs:restriction base="xs:float"/>
  </xs:simpleType>
  <xs:simpleType name="TipoNodo">
    <xs:sequence>
      <xs:element name="Numero" type="tns:TipoEntero"/>
      <xs:element name="Tipo" type="tns:TipoEntero"/>
      <xs:element name="EnEstado" type="tns:TipoNuevo"/>
      <xs:element name="Estado" type="tns:TipoNuevo"/>
      <xs:element name="Capacidad" type="tns:TipoFloat"/>
      <xs:element name="Informacion" type="tns:TipoInfo"/>
    </xs:sequence>
  </xs:simpleType>
  <xs:simpleType name="TipoSimulacion">
    <xs:sequence>
      <xs:element name="Info" type="tns:TipoInfo" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:simpleType>
  <xs:simpleType name="TipoRed">
    <xs:sequence>
      <xs:element name="Nodos" type="tns:TipoNodo" maxOccurs="unbounded"/>
      <xs:element name="Tiempo_Simulacion" type="tns:TipoSimulacion"/>
      <xs:element name="Tiempo_Rotulo" type="tns:TipoFloat"/>
      <xs:element name="Simulation" type="tns:TipoSimulation"/>
    </xs:sequence>
  </xs:simpleType>
</xs:schema>
```

**Figure 2.** XSD scheme representing flow in stochastic networks. This figure is elaborated by the authors, extending the initial proposal of Mercado [6].
The Table 1 shows a detailed description of the XML labels. Some of them are standard, others are user-designed.

**Table 1.** Description of XML labels.

| Label                   | Description                                                                                                                                 |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Scheme                  | It is the first one that must appear in an XML scheme and that encloses all its structure. It takes as an attribute the reference to a Space of Names located in the URL http://w3.org/2001/XMLSchema. This definition indicates that all the elements and types of information are related to this element, and they must be preceded with the prefix `xsd`. |
| Whole                   | It is defined as a simple XML element that is used by the entire variables Number, Type, Capacity and Successors. These do part of the sequence of the node type element. |
| Base restriction="xsd:" | **Whole type**
|                         | It is used to limit a type of simple element to a bounded set of integer values.                                                                 |
| Floating type           | It is defined as a simple XML element of a Floating Type that is used by the elements that are concerned. It denotes exactly floatingly real numbers.          |
| Base Restriction="xsd:" | **Floating Type**
|                         | It is used to limit a type of simple element to a bounded set of real numbers of floating point.                                                                 |
| Node type               | It is in use for defining an element of complex type on structures of information in the XML language. It contains such sub elements as number, type, distribution, parameters, capacity and successors. |
| Sequence                | It is in use for giving fulfillment to a sequence of elements in XML.                                                                               |
| Type number="tns:Whole Type" | It is defined to an element belonging to Node type, indicating the quantity of nodes that contains the network for the simulation. It is of a whole type. |
| T_E_arrivals            | It is defined to an element belonging to Node Type and identifies the type of node in the network is of external arrival, 2 if it is of internal arrival. |
| Distribution            | It is defined to a belonging element to. It corresponds to the distribution of time of service of the process. This parameter is of integer type. |
| Capacity                | It is defined to an element belonging to Node Type. It is identified as the maximum number of entities that can be attended simultaneously. It is an information of integer type. |
| Successors              | It is defined to an element of belonging to the element Node Type and identifies the nodes successors to the current one. |
| Successor Type          | It is in use for defining a structure of information in the XML language. Its name is Successor Type, in turn this complex element contains such sub elements as Node Numbers and Probability. |
| Node Number             | This label is defined to an element belonging to the Successor type structure. It identifies the number of the successor of the current node in the network, it is of integer type. |
| Probability             | It is defined to an element belonging to the element Successor Type, this element shows the probability of going to the successor. This information is of floating type. |
| Time                    | It is used to limit a type of simple element that it represents the time of simulation. It is a real value. |

3.2. Example of implementation

In the Figure 3(a), a graph of processes networks is shown, with relevant information to each process. The XML definition that adjusts to the structure previously presented can be seen in the Figure 3(b).
Observe that the definition of a processes network is intelligible for a user with basic knowledge of operations research.

![Diagram of processes network](image)

**Figure 3.** (a). Example of processes network. This figure is elaborated by the authors. (b). XML specification of the example network. This figure is elaborated by the authors.

4. Validation

To establish the validity of the proposed formats, was designed and used software that indicates if the XML definition is well established, and in addition, it simulates processes network to show its principal statistical indicators. Nevertheless, a simulator installed in a web server also might provide the service of simulating the network and validate it. The development of this application is constituted by the use of the language of programming Java endorsed by bookshops SAX and JDOM classes, which are used for the treatment of the information coded with XML.

Besides, another element of validation considered is the parser named Xerces, who takes charge of the validation of the model coded with XML, finally to all this set of packages of programming was added SAX, an interface with specific purposes and different from the API JDOM for the processing of information linked to an XML file.

4.1. Evidence in the validator

An XML validator represented networks process developed by Fan et al [12] and adapted later to examine the validity of incorporating new labels by Mercado [6]. This validator is in the ProcessWeb packages inside which there is an XML validator, whose content is showed in the following UML class diagram and packages at the Figure 4. The purpose of this bookshop is to validate the correct of a specification of network process, in view of the main file XSD.

4.2. Validation by means of software

A simulator has been developed based on discrete events to process the network, which has a sub package main that contains the necessary classes to carry out the simulation process. These classes are described in the Table 2.

Once checked the syntax of the file that represents the processes network and simulates the model, there is the option to see the simulation event-to-event in order to observe the conformation of the LEP...
and the trace of simulation. The actual event, time simulation and the list pending events can be appreciated in Figure 5.

**Figure 4.** UML Class diagram of the validator. This figure is adapted from Mercado [6]

**Table 2.** Description of the classes of the developed simulator. Elaborated by the authors.

| Types   | Description                                                                 |
|---------|-----------------------------------------------------------------------------|
| Node    | A type that handles the information of the nodes later to carry out the simulation and visualization of results. |
| Event   | A type used by a simulator handling the behavior of the nodes of the network as events and following a chronological order in a tidy list. Its attributes are: time of occurrence of the event, node where there happens the event, type of event (arrival or exit), and prompter to the next event in the List of Pending Events (LEP). |
| Red     | A type that takes charge encasing all the nodes of the process network in an array. The simulation begins when this arrangement is crossed and there is in use the information contained in each of the node objects stored at the array. |
| Time    | It is used to measure duration of any simulation, and its use is demonstrated in the visualization of the result of the same one. Its attributes are: hours (h), minutes (m), second (s), c. |
| Simulator | Principal type simulator. All the processes of reading of the XML file and the use of all its information. Then it sends a statistical report in case the simulation has been successful. |

**Figure 5.** Trace of events in a simulation.

In the Figure 5 appears the simulated time, the current event that is processed at simulation time and future events that will be processed, indicating the nodes where occurrence will take place. In case
of ending the simulation, is generated a statistical report of the waiting times in nodes, resource usage, service time and average size of the queue.

5. Conclusions

The definition of processes networks in XML offers many advantages as for the standardization of information and the possibility that they could be analyzed, and the models executed under applications that are employed in client-mode or server-mode across Internet. The above mentioned increases the portability among different applications with similar policies of development.

Since that the specification is functional the models can be designed and utilized in web-based simulation, which allows an access to the academic community that wants to experiment with models of this type without having to use a native format of an application. A potentially relevant aspect consists of the fact that this specification may be scalable and extensible, so that it could incorporate more information associated with the system representation.

References

[1] Banks J, Carson J, Nelson B and Nicol D 1999 Discrete-event system simulation (New Jersey: Prentice Hall)
[2] Young M 2000 XML step by step (Washington: Microsoft Press)
[3] Law, A and Kelton, D 2000 Simulation modeling and analysis (Madrid: McGraw-Hill)
[4] Guasch A, Piera M, Casanovas J and Figueras Y 2004 Modelado y simulación: Aplicación a procesos logísticos de fabricación y servicios (Barcelona: Universitat Politècnica de Catalunya)
[5] Bradley, Neil 2003 The XML schema companion 1st edition (USA: Addison-Wesley Professional)
[6] Mercado L 2011 Diseño de modelos basados en redes de procesos, utilizando el lenguaje de demarcación extensible XML (Pamplona: Universidad de Pamplona)
[7] Fernández A 2000 Estudios turísticos 146 71-82
[8] Barbosa S D F F and Marin H D F 2009 Revista Latinoamericana de Enfermagem 17 7-13
[9] Zeigler B, Praehofer H and Kim T 1990 Theory of modeling and simulation (USA: Academic Press)
[10] Kokkonen T, Jolma A and Koivusalo H 2003 Environmental Modelling & Software 18 463–71
[11] Jea K, Chang T and Cheng C 2011 Computer Standards and Interfaces 33 280-91
[12] Fan H, Ma Z, Wang D and Liu J 2018 Information Sciences 453 1-20