Computers & Geosciences 93 (2016) 63–69

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Case study

Rock.XML – Towards a library of rock physics models

Erling Hugo Jensen [a,b,*], Ragnar Hauge [c], Marit Ulvmoen [c], Tor Arne Johansen [a,d], Åsmund Drottning [e]

[a] Department of Earth Science, University of Bergen, Bergen, Norway
[b] Rock Physics Technology, Bergen, Norway
[c] Norwegian Computing Center, Oslo, Norway
[d] The University Centre in Svalbard, Norway
[e] NORSAR, Kjeller, Norway

ARTICLE INFO

Article history:
Received 22 September 2015
Received in revised form
21 April 2016
Accepted 22 April 2016
Available online 23 April 2016

Keywords:
Rock physics models
Reservoir geophysics
XML

ABSTRACT

Rock physics modelling provides tools for correlating physical properties of rocks and their constituents to the geophysical observations we measure on a larger scale. Many different theoretical and empirical models exist, to cover the range of different types of rocks. However, upon reviewing these, we see that they are all built around a few main concepts. Based on this observation, we propose a format for digitally storing the specifications for rock physics models which we have named Rock.XML. It does not only contain data about the various constituents, but also the theories and how they are used to combine these building blocks to make a representative model for a particular rock. The format is based on the Extensible Markup Language XML, making it flexible enough to handle complex models as well as scalable towards extending it with new theories and models. This technology has great advantages as far as documenting and exchanging models in an unambiguous way between people and between software. Rock.XML can become a platform for creating a library of rock physics models; making them more accessible to everyone.

1. Introduction

Rocks are solid aggregates of minerals, with possible traces of organic material and various porosities. This void space is filled with fluid or mixtures of different fluids; such as brine, gas and oil. Rock properties vary greatly due to the large number of possible constituents and compositions. For example, some rocks are unconsolidated and very loose, while others are more dense, consolidated and cemented. Even details such as if the cement is coating the grains or located at the contact points between the grains, will affect the stiffness of a rock (Dvorkin et al., 1994).

There is a wide range of theories for modelling all these different types of rocks and a review of some of them is given by Avseth et al. (2010). However, new ones and new ways of combining old theories are continuously emerging. But all modelling typically share a common workflow:

1. Identify representative solid and fluid constituents.
2. Specify constituent properties, either
   a. from tabulated values found in the literature, or
   b. calculate them
3. Calculate the effective mineral properties by combining the solid constituents.
4. Calculate the effective fluid properties by combining the fluid constituents.
5. Calculate the porous dry rock properties for a given porosity.
6. Calculate the effective fluid saturated rock properties for that porosity.

Some variations can of course occur, e.g. steps 5 and 6 are sometimes done simultaneously. Rock physics theories are applied for the various calculation steps, and the choice of theories is closely linked to the type of rock we have. Hence, interpretation in reservoir geophysics starts when choosing which rock physics model and theories to use.

We propose a format for uniquely specifying rock physics models. Hence, we can digitally store data about the various rock constituents and other relevant parameters. In addition, we store details on how the rock is composed, i.e. how to do the actual modelling by specifying which theories to use and how they should be combined. We have named this format Rock.XML and it is based on the extensible markup language XML. We have chosen

[*] Correspondence to: Løvåsbakken 21, H0103, 5145 Fyllingsdalen, Norway.
E-mail address: sschwss@hotmail.com (E.H. Jensen).

http://dx.doi.org/10.1016/j.cageo.2016.04.011
0098-3004/© 2016 Elsevier Ltd. All rights reserved.

the XML platform because it has a simple set of basic rules, yet a high degree of flexibility which makes it very easy to define custom markup languages for storing data. Furthermore, it supports a hierarchical structure which we actively use in the specification of the rock composition. (See Learning XML by Ray (2003) or XML in a Nutshell by Harold and Means (2004) for more details about XML).

XML was created as a counterpart to the Hypertext markup language (HTML). While HTML focuses on how to present data, XML focuses on what the data is (Ray, 2003). The use of XML within geoscience is not new. For example, Houlding (2001) discussed the potential for using it to store data, and Mello and Xu (2006) explain how XML can improve the efficiency of application development in geosciences. It is also possible to use XML to automate the transformation of data between different geoscience data standards (Nance and Hay, 2005). A review of the application of the geography markup language (GML) within the field of geology was given by Lake (2005), and Sen and Duffy (2005) propose another markup language which extends GML to have a wider geoscience scope. Another extension to GML was designed by Nativi et al. (2005) to integrate the use of the Network Common Data Form (netCDF). Babaie and Babaei (2005) discuss the use of XML Schema and namespaces in modelling geological objects. An online reservoir modelling software, which makes use of XML-based data handling procedures, has been created by Victorine et al. (2005). XML has also been used to describe details about rock mechanics tests such as uniaxial compression experiments (Li et al., 2012), i.e. details about the rock materials, test preparations and procedures. But we believe this is the first time it is used for storing details about rock physics models.

There are several applications for Rock.XML. It can be used in documenting applied models and specifications for a particular modelling process. This is useful not only to the individual researcher, but also for other researchers. It simplifies communication and assures collaborating colleagues are working on the same models throughout a project. The same is the case when publishing results in a journal; the relevant models can be available in Rock.XML file format as downloadable extra material – just as we are doing for this paper. Furthermore, it is possible to create software algorithms which read the files, parses the content and performs the specified modelling directly. This opens up several possibilities for modelling software such as loading and storing models to files, importing/exporting models between other software and initiating different types of batch jobs. It can be used in forward and inverse rock physics modelling, e.g. for generating constraint cubes (Johansen et al., 2013) or for generating rock physics models for use in seismic inversion (Kolbjørnsen et al., 2013) The close connection between XML and the World Wide Web invites the use of these files in various ways through browser based user interfaces or even so-called smart phone apps. It would be possible to create a library of rock physics models which are accessible online and updated with new ones as soon as they are proposed.

In this paper, we focus on the philosophy and general idea behind Rock.XML. Therefore, we show only a few examples with XML code to illustrate how it can be implemented, and instead use flowcharts to communicate our philosophy. We have included three modelling examples; for an unconsolidated friable-sand, a cemented sandstone and finally an example for a consolidated sandstone using an inclusion based model. Rock.XML files for the three rock physics models, together with a manual for the Rock.XML markup language can be downloaded as extra materials.

2. Basics of the Rock.XML format

The XML format has a set of simple rules built around the concept of defining elements. An element, identified by a tag name and with possible associated attributes, can contain data and/or other sub elements; i.e. the possibility of creating nested structures. This nesting is quite useful in rock physics modelling, where a saturated rock is often built from fluids and solids, which again may be created from mixing more fundamental constituents. The XML format is well suited for storing such recursive patterns.

The fundamental building block in an XML file is the tag, with its corresponding closing tag. Between these, there may be other tags, or element values. Consider the following example:

```
< variable >
  < label > pore pressure < /label >
  < value > 10 < /value >
< /variable >
```

Here we have the two tags “label” and “value”, which both have immediate element values. They are grouped together under a “variable” tag, to mark that they belong together. This simple, yet very flexible structure is an ideal platform to define various markup languages tailored to specific tasks or problems. We refer to our markup language as Rock.XML. The language is defined by the tags it accepts, and how these relate hierarchically to each other.

The 6-step modelling workflow provided in the introduction, shows that rock physics modelling is a process of defining building blocks and applying theories to combine them. The result can either be a new building block or a representation of the effective rock. As such, we have defined Rock.XML around the concept of constituents, elements that either may be an end result, or a part of a more complex model. We use the following constituents: *solid*, *fluid*, *dry rock* and *rock*, which becomes our basic tags. In a programming setting, these can be considered as objects with associated attributes.

A block is the set of tags and values between a tag and its closing tag. In each constituent block there is a `< label >` element, giving a unique identification (or instance name to an object) for referring to them in multiple places in the modelling, and help with the understanding of what is being defined. Furthermore, there is also always a theory block, giving a theory that defines the constituent, and its parameters. We will illustrate this by giving simple examples of our constituents.

2.1. The solid constituent

A *solid* has three associated attributes; namely bulk modulus, shear modulus and density. They are the relevant mineral properties for the modelling we want to do. This means that the information given for a solid must tell us how to compute these. Below is an example where we specify the properties for quartz (Mavko et al., 2009).

```
< solid >
  < label > quartz < /label >
  < tabulated >
    < bulk-modulus > 37 < /bulk-modulus > < !- GPa - >
    < shear-modulus > 44 < /shear-modulus > < !- GPa - >
    < density > 2.65 < /density > < !- g/cm3- >
  < /tabulated >
< /solid >
```

Here we have used `< tabulated >` as our theory. This is in reality no theory as such, but a method to set each of the attributes individually; either by value as is done here, or reference to a variable which in turn could be a result of some type of modelling. The `< !- text - >` is how comments are written in XML.

2.2. The fluid constituent

A *fluid* is a special case of a *solid*, where the shear modulus is set