Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Unia Europejska Europejski Fundusz Społeczny

# OGF DFDL (Data Format Description Language) – co to jest, do czego służy. Podstawy składni, przykłady użycia, istniejące implementacje.

Autor:

Marcin Rzewuski

# Spis treści

1. Wprowadzenie. ..... 3.
2. Historia języka..... 6.
3. Podstawy składni przykłady użycia. ..... 7.
4. Implementacje. ..... 10.
5. Zakończenie ..... 12.
6. Bibliografia ..... 12.

# 1. Wprowadzenie.

## 1.1 Czym jest język Data Format Description Language

Cytat 1:

"*What is Data Format Description Language?*
*Data Format Description Language or DFDL is pronounced like the flower 'daffodil'. It is a language designed to describe the format of data. Specifically, it is designed to describe the format of data in a way that is independent of the format itself. The idea is that you choose an appropriate data representation for an application based on its needs and then describe the format using DFDL so that multiple programs can directly interchange the described data. That is, DFDL is not a format for data; it is a way of describing any data format.*
*DFDL is intended for data commonly found in scientific and numeric computations, as well as record-oriented representations found in commercial data processing. DFDL can be used to describe legacy data files, to simplify transfer of data across domains without requiring global standard formats, or to allow third-party tools to easily access multiple formats.*
*DFDL is designed to provide flexibility and also permit implementations that achieve very high levels of performance. DFDL descriptions are separable and native applications do not need to use DFDL libraries to parse their data formats. DFDL parsers can also be highly efficient. The DFDL language is designed to permit implementations that use lazy evaluation of formats and to support seekable, random access to data. The following goals can be achieved by DFDL Note that DFDL is specifically not intended to be used to describe XML, which*
*already has well-defined ways to describe it. However, the DFDL language is built upon many of the principals of XML and is designed to make XML tooling available for use with non-XML data. Schema – A model, a framework, a plan. We will be using the generic term schema to mean a model of some data.*
*XML Schema or XSD – A model used specifically to describe the structure and content of XML data/instance documents.*
*Instance Document – A term that describes the entire stream of data that*
*we are processing in this 'run' or 'instance'. DFDL Schema – A model used specifically to describe the structure and content of non-XML data/instance document. A DFDL schema uses a subset of XML schema construct s, so a DFDL schema is actually a well-formed and valid XML schema document. How can a DFDL schema (which itself is an XML schema) describe non-XML data? The answer is 'annotations'. "* - •
**OGF DFDL home page https://www.ogf.org/ogf/doku.php/standards/dfdl/dfdl Lesson 1**

DFDL (z języka angielskiego Data Format Description Language czyli Format danych Opisu Języka) jest to język stworzony w celu opisania formatu danych. W szczególności, jest on przeznaczony do opisania danych w sposób, który jest niezależny od samego wzorca tych danych. Chodzi o to, aby wybrać odpowiednią reprezentację danych dla aplikacji w oparciu o jej potrzeby, a następnie opisać dane za pomocą DFDL tak, że wiele programów może bezpośrednio ze sobą wymieniać opisane dane. Oznacza to, że DFDL jest sposobem opisania dowolnego sposobu reprezentacji danych.
DFDL jest przeznaczony dla danych spotykanych w obliczeniach numerycznych i naukowych, jak również dla różnego rodzaju firm zajmujących się przetwarzaniem danych. DFDL może być również

używany do opisania plików ze starszymi danymi, w celu uproszczenia transferu danych między domenami bez konieczności globalnego standaryzowania tych, że danych, lub umożliwić narzędzia dla użytkowników na łatwy dostęp do tych danych. DFDL ma na celu zapewnienie elastyczności, a także pozwala na implementacje, które osiągają bardzo wysoki poziom wykonania. Opisy DFDL jest opisem naturalnym i aplikacje nie muszą korzystać z bibliotek DFDL aby analizować dany format danych.

## 1.2 Dlaczego powstał język DFDL?

Cytat 2.

***„What is DFDL?***

*DFDL is a language for describing data formats. A DFDL description allows data to be read from its native format and to be presented as an instance of an information set or indeed converted to the corresponding XML document. DFDL also allows data to be taken from an instance of an information set and written out to its native format. DFDL achieves this by leveraging W3C XML Schema Definition Language (XSDL) 1.0. [XSDL] An XML schema is written for the logical model of the data. The schema is augmented with special DFDL annotations. These annotations are used to describe the native representation of the data. This is an established approach that is already being used today in commercial systems."*

## "Data Format Description Language (DFDL) v1.0Specification" -

Michael J Beckerle, Tresys Technology Stephen M Hanson, IBM September 2014

Cytat 3.

*"Many people ask why DFDL is needed in an era where there are so many standard data formats available (e.g., why not just use XML?). There are a number of social phenomena in the way software is developed which have lead to the current situation where DFDL is needed to standardize description of diverse data formats.*

*First, programs are very often written speculatively, that is, without any advance understanding of how important they will become. Appropriately given this situation, little effort is expended on data formats since it remains easier to program the I/O in the most straightforward way possible given the programming tools in use. Even something as simple as using an XML-based data format is harder than simply using the native I/O libraries of a programming language.*

*At some point however, it is realized that the program is important because either lots of people are using it, or it has become important for business or organizational needs to start using it in larger scale deployments. At that point it is often too late to go back and change the data formats. For example, there may be real or perceived business costs to delaying a deployment of a program for a rewrite just to change the data formats, particularly if such rewriting will reduce performance of the program and increase costs of deployment. (It takes* longer to program, *but at least it's* slower *when you are done ;-)*

*Additionally, the need for data format standardization for interchange with other software may not even be clear at the point where a program first becomes 'important'. Eventually, however, the need for data interchange with the program becomes apparent. At that point, you look back at the data format and maybe it's not too complex yet. So you don't re-engineer it. But add a year or so of evolution of the software with the attendant changes here and there to the data formats and suddenly you have a real problem.*

*The above phenomena are not something that is going away any time soon. There are of course efforts to much more smoothly integrate standardized data format handling (e.g, XML) into programming languages. But it is very unclear whether these will catch on, and there is, regardless, a role for DFDL since it allows after-the-fact description of a data format.*

*DFDL is also needed for performance reasons. At the hairy edge of computing people are always trying to process ever more data to gain some competitive advantage. At this edge, the performance penalty from using verbose data formats like XML can become very burdensome. DFDL can be used to describe densely packed bit-optimized formats" -*

http://cboblog.typepad.com/cboblog/2008/07/dfdl-data-forma.html
blog Mike Beckerle
DFDL = Data Format Description Language - The syntax of data

W świecie, w którym przepływ informacji ma coraz duże znaczenie, jak również konieczność przetwarzania coraz większej ilości danych zaistniała konieczność opracowania standardu opisu danych, który mógłby być wykorzystany w sposób uniwersalny przez różnego rodzaju aplikacje. Programy są bardzo często zapisywane w taki sposób, w którym nie zwraca się uwagi na format danych, programuje się tak aby jak najprościej uzyskać zamierzony efekt. Nawet coś tak prostego jak użycie formatu danych XML jest trudniejsze niż po prostu zastosowanie formatów rodzimych I / O bibliotek danego języka programowania. W pewnym momencie jednak zauważamy, że format danych który zastosowaliśmy nie spełnia wymagań, których byśmy oczekiwali. W tym miejscu często jest już za późno, aby wrócić i zmienić. (Na przykład, dane mogą być nieprawdziwe lub koszty przeprowadzenia danej operacji mogą być duże. )

DFDL potrzebne jest również ze względu na wydajność. W obecnych czasach ludzie starają się przetwarzać coraz większą ilość danych, aby zdobyć przewagę nad konkurencją. Na tym polu, spadek wydajności z użyciem rozwlekłych formatów danych XML może być bardzo uciążliwy. DFDL może być stosowane w celu opisania różnego formatu danych w celu ich optymalizacji. DFDL pozwala na natywny opis różnego rodzaju danych.

# 2. Historia języka.

Cytat 4

*"DFDL was created in response to a need for grid APIs to be able to understand data regardless of source. A language was needed capable of modeling a wide variety of existing text and binary data formats. A working group was established at the Global Grid Forum (which later became the Open Grid Forum) in 2003 to create a specification for such a language.*

*A decision was made early on to base the language on a subset of W3C XML Schema, using <xs:appinfo> annotations to carry the extra information necessary to describe non-XML physical representations. This is an established approach that is already being used today in commercial systems. DFDL takes this approach and evolves it into an open standard capable of describing many text or binary data formats.*

*Work continued on the language, resulting in the publication of a DFDL 1.0 specification as OGF Proposed Recommendation GFD.174 in January 2011. The latest revision is GFD.207 published in November 2014 which obsoletes GFD.174 and incorporates all issues noted to date (also available as html). A summary of DFDL and its features is available at the OGF. Any issues with the specification are being tracked using Redmine issue trackers" -*

Wikipedia, Data Format Description Language

DFDL został stworzony w odpowiedzi na potrzeby gridu API, aby można było przetwarzać dane niezależnie od źródła. Język był potrzebny dla obsługi różnorodnych tekstowych i binarnych formatów danych. Zespół odpowiedzialny za stworzenie specyfikacji tego języka powstał w 2003 roku przy Global Grid Forum (który później stał się Open Grid Forum ) Na początku podjęto decyzję aby ten język oprzeć na schemacie W3C XML, używając <xs: appinfo> . DFDL przyjmuje to podejście i rozwija go do otwartego standardu zdolnego do opisywania wielu tekstowych czy binarnych formatów danych. W styczniu 2011 roku OGF opublikowało specyfikacje języka w doku męcie DFDL 1.0. Ostatnią wersję języka obejmuje dokument GFD.207 opublikowany w listopadzie 2014 roku, która obejmuje wszystkie aktualne zagadnienia związane z tym językiem.

# 3. Podstawy składni przykłady użycia.

Rozważmy następujący schemat danych zapisanych w XML

```
<w>5</w>
<x>7839372</x>
<y>8.6E-200</y>
<z>-7.1E8</z>
```

Logiczny opis tych danych możemy zawrzeć w następującym fragmencie XML:

```
<xs:complexType name="example1">
<xs:sequence>
<xs:element name="w" type="xs:int"/>
<xs:element name="x" type="xs:int"/>
<xs:element name="y" type="xs:double"/>
<xs:element name="z" type="xs:float"/>
</xs:sequence>
</xs:complexType>
```

Teraz załóżmy, że mamy te same dane, ale nie reprezentowane w formacie XML Binarne dane mogą być zapisane w postaci szesnastkowej.

```
0000 0005 0077 9e8c
169a 54dd 0a1b 4a3f
ce29 46f6
```

Aby opisać to w języku DFDL, bierzemy nasz dokument w schemcie XML, w którym oryginalny model danych opiszemy i zdefiniujemy w następujący sposób:

```
<xs:complexType>
 <xs:sequence>
  <xs:element name="w" type="xs:int">
   <xs:annotation>
    <xs:appinfo source="http://www.ogf.org/dfdl/">
     <dfdl:element representation="binary"
                   binaryNumberRep="binary"
                   byteOrder="bigEndian"
                   lengthKind="implicit"/>
    </xs:appinfo>
  </xs:annotation>
 </xs:element>
 <xs:element name="x" type="xs:int ">
   <xs:annotation>
    <xs:appinfo source="http://www.ogf.org/dfdl/">
     <dfdl:element representation="binary"
           binaryNumberRep="binary"
           byteOrder="bigEndian"
           lengthKind="implicit"/>
   </xs:appinfo>
  </xs:annotation>
 </xs:element>
 <xs:element name="y" type="xs:double">
   <xs:annotation>
     <xs:appinfo source="http://www.ogf.org/dfdl/">
```

```
      <dfdl:element representation="binary"
            binaryFloatRep="ieee"
            byteOrder="bigEndian"
            lengthKind="implicit"/>
    </xs:appinfo>
    </xs:annotation>
  </xs:element>
   <xs:element name="z" type="xs:float" >
     <xs:annotation>
      <xs:appinfo source="http://www.ogf.org/dfdl/">
       <dfdl:element representation="binary"
            byteOrder="bigEndian"
            lengthKind="implicit"
            binaryFloatRep="ieee" />
   </xs:appinfo>
  </xs:annotation>
 </xs:element>
</xs:sequence>
</xs:complexType>
```

W ten prosty sposób dane możemy wyrazić w postaci binarnej.

Te same dane możemy opisać korzystając z kodowania UTF-8 z wykorzystaniem separatora pól (przecinek) i separatora dziesiętnego (kropka):

```
<xs:complexType>
 <xs:sequence>
  <xs:annotation>
   <xs:appinfo source="http://www.ogf.org/dfdl/">
    <dfdl:sequence encoding="UTF-8" separator="," />
   </xs:appinfo>
  </xs:annotation>
  <xs:element name="w" type="xs:int">
   <xs:annotation>
    <xs:appinfo source="http://www.ogf.org/dfdl/">
     <dfdl:element representation="text"
               encoding="UTF-8"
               textNumberRep ="standard"
               textNumberPattern="####0"
               textStandardDecimalSeparator="."
               lengthKind="delimited"/>
     </xs:appinfo>
    </xs:annotation>
</xs:element>
<xs:element name="x" type="xs:int">
   <xs:annotation>
    <xs:appinfo source="http://www.ogf.org/dfdl/">
      <dfdl:element representation="text"
            encoding="UTF-8"
            textNumberRep ="standard"
            textNumberPattern="########0"
            textStandardDecimalSeparator="."
            lengthKind="delimited"/>
   </xs:appinfo>
  </xs:annotation>
</xs:element>
<xs:element name="y" type="xs:double">
   <xs:annotation>
    <xs:appinfo source="http://www.ogf.org/dfdl/">
     <dfdl:element representation="text"
              encoding="UTF-8"
              textNumberRep ="standard"
              textNumberPattern="0.0E+000"
```

```
                textStandardDecimalSeparator="."
                lengthKind="delimited"/>
     </xs:appinfo>
    </xs:annotation>
</xs:element>
<xs:element name="z" type="xs:float">
    <xs:annotation>
     <xs:appinfo source="http://www.ogf.org/dfdl/">
      <dfdl:element representation="text"
          encoding="UTF-8"
          textNumberRep ="standard"
          textNumberPattern="0.0E0"
          textStandardDecimalSeparator="."
          lengthKind="delimited"/>
    </xs:appinfo>
   </xs:annotation>
 </xs:element>
 </xs:sequence>
</xs:complexType>
```

Przykład pochodzi z opracowania:

## "Data Format Description Language (DFDL) v1.0Specification" -

Michael J Beckerle, Tresys Technology  Stephen M Hanson, IBM  September 2017

# 4. Implementacje.

Cytat 5

"*Implementations of DFDL processors that can parse and serialize data using DFDL schemas are available.*

- *IBM has a production-ready DFDL 1.0 streaming parser, modeler and visual tester. This is available in several IBM products including IBM Integration Bus (formerly known as IBM WebSphere Message Broker). A free developer edition is available.*
- *An Open Source DFDL processor known as Daffodil is under active development with a parser available from spring 2015.*
- *European Space Agency project S2G Data Viewer includes a parser DFDL4S that implements a subset of the DFDL 1.0 specification.*

*A presentation is available that describes IBM DFDL and Daffodil.*

*A public repository for DFDL schemas that describe commercial and scientific data formats has been established on GitHub. DFDL schemas for formats like UN/EDIFACT, NACHA, HL7 and ISO8583 are available for free download."*

Wikipedia, Data Format Description Language

Implementacje systemu DFDL, za pomocą którego można analizować i szeregować dane za pomocą schematów DFDL :
• IBM ma gotowe rozwiązania wykorzystujące DFDL 1.0 produkcji strumieniowe parser, 1.0 i tester Modeler wizualny. Jest dostępny w kilku produktów IBM, *IBM Integration Bus (* (dawniej IBM WebSphere Message Broker).
• Projektr Open Source DFDL znany jako *„Daffodil"* od wiosny 2015
• projekt Europejskiej Agencji Kosmicznej S2G Data Viewer zawiera DFDL4S ,który implementuje podzbiór 1.0 specyfikacji DFDL.

Cytat 6

*"Daffodil is the open source implementation of a parser for the Data Format Description Language (DFDL), a specification created by the Open Grid Forum. DFDL is capable of describing many data formats, including textual and binary, commercial record-oriented, scientific and numeric, modern and legacy, and many industry standards. It leverages XML technology and concepts, using a subset of W3C XML schema type system and annotations to describe such data. Daffodil uses this description to parse data into an XML infoset for ingestion and validation."*

http://opensource.ncsa.illinois.edu/confluence/display/DFDL/Daffodil%3A+Open+Source+DFDL

Daffodil: Open Source DFDL

Created by Rob Kooper, last modified by Stephen Lawrence on Mar 16, 2015

Projektr Open Source DFDL znany jako *„Daffodil"* otwarta implementacja dla języka DFDL.

# 5. Zakończenie

Język DFDL jest zdolny do opisywania wielu formatów danych, w tym wiele standardów branżowych wykorzystywanych w różnego rodzaju instytucjach czy to komercyjnych czy naukowych. Wykorzystuje technologię XML . Język ten ma coraz większe znaczenie w świecie, w którym coraz szybsze przetwarzanie danych jest niezbędne .i ma znaczący wpływ na funkcjonowanie gospodarki.

# Bibliografia

1. "Data Format Description Language (DFDL) v1.0Specification" - Michael J Beckerle, Tresys Technology Stephen M Hanson, IBM September 2014
2. http://cboblog.typepad.com/cboblog/2008/07/dfdl-data-forma.html
   blog Mike Beckerle
   DFDL = Data Format Description Language - The syntax of data
3. Wikipedia, Data Format Description Language
4. http://opensource.ncsa.illinois.edu/confluence/display/DFDL/Daffodil%3A+Open+Source+DFDL
   Daffodil: Open Source DFDL
   Created by Rob Kooper, last modified by Stephen Lawrence on Mar 16, 2015
5. https://www.ogf.org/ogf/doku.php/standards/dfdl/dfdl Lesson 1
   OGF DFDL home page