ADF – AGRICULTURE DATA FORMAT V1

MATTEO NICOLI

1. Introduction

Questo documento definisce le specifichwe del formato *Agriculture Data Format* (from now on ADF).

I concetti fondamentali riguardanti questa struttura dati sono i seguenti:

- L'archiviazione in formato ADF si basa sulla periodicità dei dati. Questo significa che i dati sono strutturati in serie che idealmente reppresentano misurazioni effettuate ad intervaalli arbitrari e regolari.
- Deve essere possibile eseguire operazioni di push di un elemento in fondo alla lista delle iterazioni, e pull dell'ultimo elemento inserito. Questo perchè, uno degli scopi fondamentali di questo standard, non è solo quello di archiviare e organizzare i dati già raccolti, ma anche quello di collezionarne di nuovi.

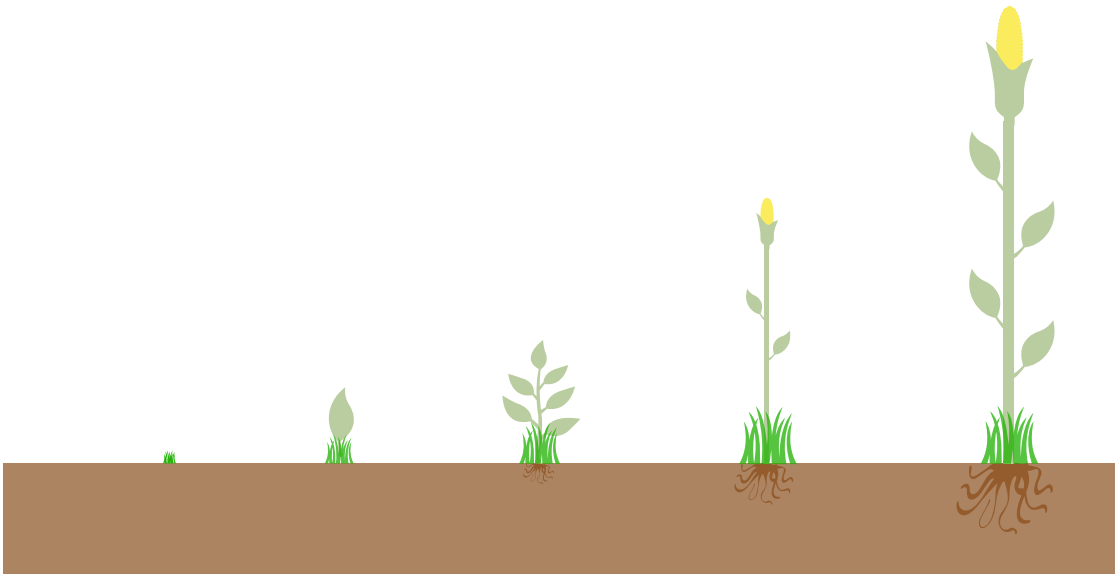

Figure 1. An (over)simplified schema of a growing crop. It is useful to show how we are encoding the information concerning series and measurements.

2. Data representation

All data structures that require more than one byte to be represented, are stored, in network byte order, i.e. big-endian, where most significant bytes come first. Dove non espressamente specificato, si intendono gli integer come unsigned -byte integers, and floating point numbers as

Date: April 23, 2024.

IEEE 754 32-bit base-2 floating-point numbers. Here is a list of the abbreviation used in the following pages.

code	description
UI	1-byte unsigned integer
UI2	2-byte unsigned integer
UI4	4-byte unsigned integer
F	IEEE 754 4-byte base-2 floating-point
ADD_T	6-byte structure composed by UI2 + F

TABLE 1

For what concerns the structure, ADF is composed by a 32-byte header (ADFHDR), a metadata chunk (ADFMTA), and possibly n series (ADFSRS), with $n \geq 0$. L'header contiene tutti i metadati costanti, che non possono dunque cambiare a seguito di un update, mentre il metadata chunk contiene i metadati variabili. Possiamo osservare nello schema in Figura 2

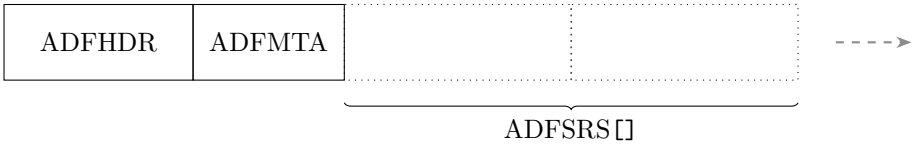

FIGURE 2. This diagram shows the structure of an ADF file. The first two chunks starting from the left are drawn with solid border because they are mandatory, while the entire series section is composed by dotted chunks, because they may not be present.

All the series chunks have the same length. La ragione di questo è che ciascuna serie rappresenta un *intervallo periodico*. Per farsi un'idea abbastanza intuitiva di questo, possiamo prendere come riferimento i dati agricoli in una qualunque delle descrizioni informali nelle quali sono espressi. Diremo di più su questo nella Sezione 2.3.

2.1. **ADFHDR – ADF Header.** Each ADF file contains a 24-byte header, divided as shown in Table 2. The header contains all those data that remains immatable through all the series recorded.

chunk	type	content
s	UI4	signature
v	UI	version
ft	UI	farming technique
n_{wl}	UI4	number of wavelengths
$\min(W_l)$	UI4	minimum wavelength in nanometers
$\max(W_l)$	UI4	maximum wavelength in nanometers
n_c	UI4	number of chunks
crc_header	UI2	CRC-16 calculated on header bytes

TABLE 2

Signature. The first four bytes of any ADF file always contain the four (hexadecimal) bytes 0x40 0x41 0x44 0x46, that corresponds to the string "`@ADF`".
Version. Represented as a 1 byte usigned integer. Versions are numbered progressively, without any distinction between major and minor releases.
Farming technique. A 1-byte unsigned integer is dedicated to encode the farming technique use to grow the crop. Allowed values are presented in Table 3. This field may contain a value in the range 0x00 – 0xFF; 0xij

0x00	Regular	0x01	Indoor	0x02	Indoor protected	0x03	Outdoor
0x10	Artificial	0x11	DDDDD	0x12	DDDDD		
0x20	Hydroponics	0x21	Anthroponics	0x22	DDDDD		
0x30	Aeroponics	0x31	Fogponics	0x32	DDDDD		

Table 3. Farming technique allowed values.

Wavelengths. Information concerning light exposure are encoded in the header with three 4-byte unsigned integers: n_{wl}, $\min(W_l)$, and $\max(W_l)$. Where $\min(W_l)$ is the lower bound of the light spectrum we are considering, $\max(W_l)$ is the upper. The spectrum between $\min(W_l)$ and $\max(W_l)$ is divided in n_{wl} intervals. This allow us to encode the spectrum with arbitrary precision.
Number of chunks. It represents the number of measurements for each series. As we saw above, each series is an iteration

2.2. **ADFMTA – ADF Metadata.** Come abbiamo anticipato nella Sezione 1, all'interno della sezione ADFMTA sono contenuti i metadati *non costanti*, come ad esempio il numero delle serie, il periodo complessivo (che dipende dal numero delle serie) e i codici degli additivi. Ogni qualvolta si aggiunge una nuova serie, bisogna aggiornare i metadati della sezione ADFMTA.

I campi contenuti nella sezione ADFMTA sono rappresentati nella Table 4 qui sotto con le relative dimesioni in byte.

chunk	type	content
n_s	UI4	number of series recorded
p	UI2	period measured in seconds
n_{add}	UI2	number of both soil and atmosphere nutrients in catalogue
$additives$	UI4 $\times$ n_{add}	An array containing the nutrients code
crc_meta	UI2	CRC-16 calculated on metdata bytes

Table 4

Number of series – $\mathbf{n_s}$. is a 4-byte unsigned integer that represents the number of series recorded in the file. This number must be incremented each time new a new series is added.
Period. is a 2-byte unsigned integer that contains information on the duration in seconds of each series. The series all have the same duration. With the period we can easily calculate the total duration of the encoded data is given by multiplying the total number of series and the period of each series.

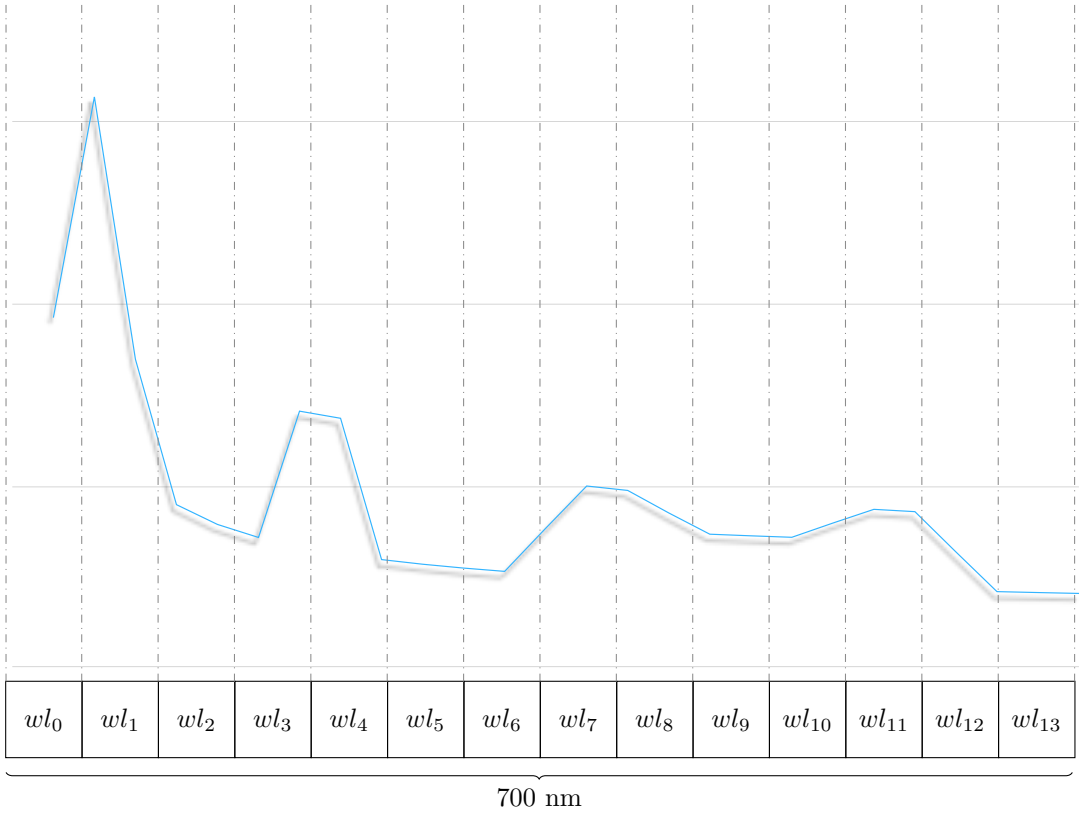

FIGURE 3. Here is an example of a wavelength spectrum with $\min(W_l) = 350$ nm, $\max(W_l) = 1,050$ nm, and $n_{wl} = 14$. Each spectrum interval represents a 50 nm range. Each wl_i contains a 4-byte floating point number that represents the light radiation measured in $\mathrm{W/m^2}$.

Additives. One of the most crucial things that this format should be able to encode are nutrients-related data. Those represents all the additives (we shall use the word "nutrients" to indicate them as well) present either in the soil or in the atmosphere. To ensure greater generality, the code for each additive is contained in the 'additives' array of the metadata; in each array there will be indices of the corresponding elements. This allows us to code the chemical elements indicated as additives with any number of bytes large, decreasing redundancy. In our case, the default code we use to designate each additive is the unique ID of the *Chemical Entities of Biological Interest* (from now on ChEBI) database. in Table 5 we can see the code of some of the most common soil nutrients.

The field n_{add} represents the size of the *additives* array, which contains a unique code of each additive. As we said above, each code represents univocally a chemical compound, with an arbitrary ID. Any time a new series is added, it may contain a reference to a new additives, that is not yet present in the array on the additives in the metadata section. In this case, the function This array *must* be updated in a conservative way, i.e. by pushing new elements at the end, whitout changing the order of the previous..

Macronutrients			Micronutrients		
ChEBI ID	formula	IUPAC name	ChEBI ID	formula	IUPAC name
17632	$[NO_3]^-$	Nitrate	29033	$[Fe]^{2+}$	Iron(2+) ion
28938	$[NH_4]^+$	Ammonium ion	29034	$[Fe]^{3+}$	Iron(3+) ion
43474	$[HPO_4]^{2-}$	Hydrogenphosphate	22908	$[BO_3]^{3-}$	Borate
39745	$[H_2PO_4]^-$	Dihydrogenphosphate	37255	$[Zn]^{2+}$	Zinc(1+) ion
29103	$[K]^+$	Potassium(1+) ion	49552	$[Cu]^+$	Copper(1+) ion
18420	$[Mg]^{2+}$	Magnesium(2+) ion	29036	$[Cu]^{2+}$	Copper(2+) ion
29108	$[Ca]^{2+}$	Calcium(2+) ion	29035	$[Mn]^{2+}$	Manganese(2+)
16189	$[SO_4]^{2-}$	Sulfate	36264	$[MoO_4]^{2-}$	Molybdate
			17996	$[Cl]^-$	Chloride

TABLE 5. A list of the 17 most common soil nutrients.

2.3. **ADFSRS – ADF Series.** As we saw in the general description in Section 1, each ADFSRS is independent of the others. In Table 6 we can see the list of the fields of each ADFSRS chunk. We shall give a more detailed description of each of them below.

chunk	type	content
i	F × n_{wl}	the energy flux of light radiation measured in W/m^2
t	F × n_c	average environmental temperature, measured in °C
w	F × n_c	water use measured in l
pH	UI	average pH of the soil
ρ	F	average density of the soil measured in kg/m^3
P	F	pressure measured in bar
n_add_soil	UI2	number of both soil nutrients
n_add_atm	UI2	number of both atmosphere nutrients
$additives_s$	ADD_T × n_add_soil	an array that contains the nutrients in the soil
$additives_a$	ADD_T × n_add_atm	an array that contains the nutrients in the atmosphere
r	UI4	indicates how many consecutive times this series is repeated
crc_series	UI2	CRC-16 calculated on each series' bytes

TABLE 6

3. OPERATIONS

Le quattro funzioni che operano sui file ADF, che devono essere presenti in ogni implementazione dello standard, sono:

We can read the details about their specification in the following paragraphs.

Each of the functions listed above *must* include in its returns datatype a 2-byte unsigned integer that represents the status code. For a list of availables status code see Section 4

3.1. **Additives.**
Reindex.

3.2. **Series.**
Add. This function adds a new series at the end of the collection. If the series you are trying to add is equal to the series

Remove.
Update.

4. Codes

Status codes are unsigned 2-byte integer. They are used as return value for

Error codes are expressed with a 2-byte unsigned integer. Table 7 contains a description of each admissible error code.

code	error	caused by
0x01u	HEADER_CORRUPTED	The *crc_header* contained in the last two bytes of the header does not match that calculated during the unmarshal of the header.
0x02u	METADATA_CORRUPTED	The *crc_meta* contained in the last two bytes of the metadata does not match that calculated during the unmarshal of the metadata.
0x03u	SERIES_CORRUPTED	The *crc_series* contained in the last two bytes of each series does not match that calculated during the unmarshal of each series.
0x04u	ZERO_REPEATED_SERIES	The field 'repeated' in a series is set to 0. Tipically this is caused during the add_series operation.
0x05u	EMPTY_SERIES	You are trying to delete a series, but there are none.
0x06u	TIME_OUT_OF_BOUND	You are trying to select a series (tipically for update), by specifying a time (in seconds) that is greater than the 'period_sec' contained in the metadata section.
0x07u	ADF_ADDITIVE_TABLE_OVERFLOW	While reindexing the additives the program is trying to resize the additives' lookup table, but it is too big. This should never happen, except for embedded systems running on very tiny chips.
0xFFFFu	RUNTIME_ERROR	It is the most generic error code. It can be caused by any operation that does not concern directly any operation performed by the user on the ADF data

Table 7