Dabatree

Wlodzimierz Holsztynski

December 28, 2016

Contents

1 Introduction

I'll define - as I have done in the past - *the daba*; and now I will also introduce the *dabatree* which is a family of dabas which are similar to the (main) daba, and all jointly form a tree of dabas (first of all the dabatree includes the daba).

NOTE *I've started to write this file on Sunday, 2016-12-25.*

2 General comments

The world is CHAOS. The world is hopeless but in a cruel way it's beautiful and allows for hope. Here and there CHAOS can be harnessed by simplicity and logic. It's a paradox–let's enjoy it (on each occasion whnever it lasts).

Thus, the *dabanese* language and *the daba* universal data base are utmost logical and simple and flexible. In particular, they have a natural ability for a self reparation.

The order in which new dabarecords appear reflect mainly the urgency of developing a working system here and now. This doesn't have much in common with logic. However, one may imagine each current dabaversion as a somewhat chaotically and ad hoc selected part of the future ultimate daba. The logical stucture is already there due to the seniority relation. The seniority relation is always the same.

It's likely that with the time passing some dabarecords are not much useful anymore, that other dabarecord might be clearly better.

Then we simply create a new dabarecord while we never remove any, never. For instance, we may have a dabadocument which is over a hundred years old. We will be able to go back to the daba state from the past, to interpret that dabadocument as at the time when it was created.

The dabatree (including the daba) and dabanese would organize the knowledge, would cure journalism, would help translations between different natural languages (via dabanese), ...

The dabanese (and the daba) would develop a child's brain dramatically better than it's done by the education these days.

3 Empty dabaphrases and singletons

There are two empty dabaphrases, namely { } and ( ) where there is nothing between the respective braces or parenthesis. They do play their role. They may serve for place holders where under a different but still similar circumstances a phrase is not empty in one instance but empty in the other one. Then, in that other (empty) case, we have a chance to indicate that on similar occasions you should expect some positive information (more than just nothing). The situation is similar to the occurences of the digit 0 , e.g. in 1024. While zero 0 stands for nothing it still plays a useful role.

We also need to pay attention to the difference between a dabaphrase and another dabaphrase, called *singleton*, for which the former one is its only direct subphrase. The difference can be dramatic also in the case of the daba, i.e. of dabarecords. But first, let's list the six possible singleton daba formats:

- { { ... } }
- { (...) }
- { [...] }
- ({ ... })
- ((...))
- ([...])

NOTE The accented dabaphrase [...] is always one of the following more detailed ways: [*dabagram*] or [{ ... }] or [{ ... }].

There are, by definition (!), exactly six different singleton formats, they are listed above. When S is a singleton then the dabaphrase [S] may also feel like a singleton but it is not–the definition is formal and rigit. The dabaphrase can be an *accented singleton* (when P is a singleton) but it is not simply a *singleton*.

Consider the following two dabaphrases: { *hmn dog cat* } and { { *hmn dog cat* } } , The former dabaphrase has three direct subphrases (namely, dabagrams of a human and a cat and a dog) while the later phrase has only one subphrase, namely the former dabaphrase is its direct subphrase.

In the case of dabarecord 2 (look for a newer note: *Dabaversion 1*) we can see the following dabaphrase: ((*Albert* [*Einstein*])). This dabaphrase is a singleton. In the context of dabarecord 3, this singleton means that there is only one name which consists of two words, *Albert Einstein* (where *[Einstein]* is accented hence *Einstein* serves as *last name*).

On the other hand, if instead of the above singleton we had only its single subphrase, i.e. (*Albert* [*Einstein*] then the meaning would be drastically different. Instead of one name we would have one name, namely *Einstein*, and one daba synonym, namely *Albert* (that would be horrible!).

4 A definition of the daba

The daba is an ordered dabalist (an ordered dabaphrase which has no accent). This already means that the daba is of the form (A B ...) where A B ... stand for dabaphrases. Moreover, these (main or direct) subphrases of the daba, called *dabarecords*, are ordered dabaphrases, so that the daba has a format of

((...) (...) (...) ...)

Thus each dabarecord is an ordered sequence of its direct subphrases called *dabafields*. Each dabarecord has at least four and a maximum of five different dabafields, namely:

- dabafield 0–*historical id*;

- dabafield 1–*the dabadate*;
- dabafield 2–*the dabaname and its dabanyms*;
- dabafield 3–*dabafamily: parents, children and relatives*;
- datafield 4–*dabalog*.

The last dabafield, dabalog, is optional.

For each dabarecord the first dabafields 0 1 2 are fixed forever while the family dabafield will change in time. Neverthless, the daba always allows to recover the full history of each dabarecord and of the entire daba. The eventual dabalog field can only grow in time, nothing will be removed from it.

The 4-field format of a dabarecord is as follows:

$$(\; n \; d \quad \{ \; [\; \ldots \;] \; \ldots \; \} \quad (\; \{ \; \ldots \; \} \; \{ \; \ldots \; \} \; \{ \; \ldots \; \} \;) \;)$$

where n is a non-negative integer (i.e. the historical id), d stands for the dabadate,, naxt field $\{ \; [\; \ldots \;] \; \ldots \; \}$ is for the dabaname and its eventual dabaysms; finally, the dabafilel 3 is an ordered list of three dabasets of parents, children and relatives (in this order). The parents/children/relatives of a dabarecord are some other dabarecords (indicated by their historical ids)..

The 5-field dabarecord has the extra dabalog field:

$$(\; n \; d \quad \{ \; [\; \ldots \;] \; \ldots \; \} \quad (\; \{ \; \ldots \; \} \; \{ \; \ldots \; \} \; \{ \; \ldots \; \} \;) \qquad (\; \ldots \;) \;)$$

4.1 Historical id (field 0)

Let me repeat just this one time that each dabarecord has several of its direct subphrases called fields. The first of these fields is the non-negative integer, called the *historical id*, starting with 0. We see the format of the daba in a greater detail:

$$(\; (\; 0 \; \ldots \;) \; (\; 1 \; \ldots \;) \; (\; 2 \; \ldots \;) \; \ldots \;)$$

so that the first three dabasymbols on the left are two left dabaparentheses followed by digit 0.

4.2 The dabadate (field 1)

The historical id is followed by a date in one of the following formats– here is an example:

- $2016-12-25$
- $(\ 2016-12-25\ \ h.02\ )$
- $(\ 2016-12-25\ \ h.18:48\ )$
- $(\ 2016-12-25\ \ h.20:48:33\ )$

A string like $2016-12-25$ includes no spaces hence it is a dabagram. Also string $h.19$ nor $h.18:48$ nor $h.18:48:33$ includes no space. Thus these three are dabagrams too.

The daba changes in time. It only grows. Nothing gets ever deleted (only the family of a record gets modified when needed; the past family history can be always easily recovered).

The dabarecords of the daba are added, one at the time, always with the consecutive historical ids. The historical ids and the dabadates increase together.

NOTE *This agreement of the historical ids and dabadates would be indeed ideal. When the work on the daba gets intensive then some dabarecords may get created in parallel. Occasional discrepancy of ids and subphrases may happen. This will be still fine as long as the iterated relation between parents and children are* ***always*** *preserved (see the respective coming material).*

Thus the daba now may look something like this:

$$
\begin{gathered}
(\\
(\ 0\ (\ 2016-02-25\ \ h.18:48\ )\ \ldots\ )\\
(\ 1\ (\ 2016-12-05\ \ h.20:48:33\ )\ \ldots\ )\\
(\ 2\ (\ 2016-12-26\ \ h.18:48\ )\ \ldots\ )\\
(\ 3\ (\ 2017-12-08\ )\ \ldots\ )\\
\ldots\ )
\end{gathered}
$$

The above 6 lines represent just one dabaphrase, actually the entire daba (the three dots in the last sixth line stand for additional

dabarecords or for none). The *newline* is treated by dabanese just as another white space. Whenever a dabaphrase includes one or more white spaces in a raw it may include any number of them instead without changing the structure of the phrase (thus the number of the consecutive white spaces do not affect of the meaning of a dabatext.

At this time the above dabadats are ficticious. When a dabadate is attached to a new record, they stay together forever–neither the historical id nor the dabadate of a record can ever change. Soon, I will start the daba for real. It will have only two records to form the initial version 1. Then the more extensive versions will be written down. The last historical id of the current daba, say id:=112, will serve as the dabaversion n. When you cut off the last existing dabarecord then you get a lower dabaversion. E.g. removing just the last dabarecord from dabaversion 112 reduces the obtaind copy to dabaversion 111.

4.3 Dabaname (field 2)

The field after the dabadate is the *dabaname*. The dabaname field is an unordered dabaphrase, i.e. a *dabaset* $\{ \dots \}$. . As a minimum, These curly braces contain at least one dabaphrase, and exactly one of them one is accented, $\{ [\dots] \dots \}$. The accented subphrase of a dataname is called the main name. Other subphrases (if any) of a dabaname are called dabasynonyms or *dabaysms* for short. Remember that a dabaphrase may be simply a dabagram. Thus a number of dabanames and dabaysms will be simply dabagrams. For instance, an initial dabaversion (to be written completely later) may look like something of this kind:

$$
\begin{gathered}
(\\
(\ 0 \ (\ 2016{-}02{-}25 \ \ h.18{:}48 \ ) \ \{ [\ \forall \ ] \ all \ \bigvee \} \ \dots \ ) \\
(\ 1 \ (\ 2016{-}12{-}05 \ \ h.20{:}48{:}33 \ ) \ \{ \ [\ \emptyset \ ] \ nic \ \bigwedge \ \} \ \dots \ ) \\
\dots \ )
\end{gathered}
$$

Here the main name for dabarecord 0 is the mathematical symbol *for all*: $\forall$, while its dabaysms are *all* and another mathematical symbol $\bigvee$. Then, for the dabarecord 1, the main dabaname is the symbol of the *empty set*: $\emptyset$, while its dabaysms are *nic* (which is a Polish equivalent of the English *nothing*). In the case of these two

dabarecords, their all dabanames and dabaysms are simply dabagrams. (The fields 0 and 1, i.e. the historical id and dabaname fields will stay for good; only the two dabadates will be fixed, again once and forever forever, when I get to the dabaversion 0).

The dabayms which appear in the dabaname field are the original dabaysms. Later, new dabaysms can be introduced in the dabalog field.

4.4 Dabafamily (dabafield 3): parents, children and relatives

The dabafamily field has format:

$$(\{ \dots \} \{ \dots \} (\dots))$$

where the first subphrase of the dabafamily field stores parents of the given dabarecord, the second stores children, and the last one stores relatives and antonyms. This last (ordered) subdabaphrase may be empty, ( ) but otherwise it must contain **two** unordered subdabaphrases, like this ({ ...} { ... }) – even if one of them is an empty unordered dabaphrase; the first of the two stores some relatives while the other one stores antonyms (relatives and antonyms are represented simply by their dabarecord ids).

Some of these lists may be empty where there is nothing between the respective pairs of parentheses (braces). This may happen on occasions in the case of relatives/antonyms. However except for the dabarecord 0 every dabarecord must have at least one parent, and except for the dabarecord 1 every dabarecord must have at least one child.

Acyclic Axiom The daba doesn't contain any sequence of dabarecords

$$r_0 \ \dots \ r_n$$

where n is any positive integer and r_k is a parent of r_{k-1} for every $k = 1 \dots n$, and r_0, at any time.

This relation to parents and childrens induces a partial order of the dabarecords (one introduces grandparents and grandchildren, etc)– when a sequence $r_1 \ \dots \ r_n$ as above satisfies the condition: r_k is a parent of r_{k-1} for every $k = 1 \dots n$ then we say that dabarecord r_n is older than dabarecord r_0. This way we get the seniority partial order.

Persistency Axiom If, at any time, a dabarecord b is older than a dabarecord a then since that time dabarecord b will stay always older than dabarecord a.

On the other hand there are no formal restrictions on selecting relatives. This is possible since relatives do not address the question of seniority.

Th seniority is a formal concept imposed strictly by parents and children. Nevertheless the daba's idea of seniority goes along the John von Neumann's style: a set is senior both to its elements and to its subsets. Thus, dabarecord 0 has the highest seniority while dabarecord 1 has the least seniority of all dabarecords.

4.5 Dated dabaphrases

A *dated dabaphrase* is an ordered dabaphrase (a pair) of the form (d p), where d a data phrase as described above in the section *The dabadate (field 1)*. And p is an arbitrary dabaphrase. Here is an example:

(2016−12−20 (∅ 1930–01–22 (*Tom Busyman*) 2012−06–14 ∅))

This perhaps tells us that the note was made on **2016-12-20** (that's the first subphrase), and the note tells us that someone named Tom Busyman was born on 1930-01-22, and has died on 2012-06-14.

In general the subphrase p of a dated dabaphrase (d p) can be a dated phrase itself. In the given example it is not despite the fact that it is an ordered dabaphrase which includes a date.

4.6 Dabalog (field 4)

The details of a description of the dabalog field will appear in a separate note. Here let's start with knowing that the dabalog has three direct subphrases so that it looks like:

(n f g)

All three subphrases n f g are ordered lists of their dated subphrases, where subphrase n stands for dabasynonyms (dabaysms), subphrase f stores family updates, and subphrase g stores the updates on other information besides dabaysms and family relations.

Each and any of these subphrases can be empty or may have just a single subphrase of its own, e.g. the dabalog which has no dabaysms would look as follows:

$$(\, (\,) \quad f \; g \,)$$

Or, instead, if the other two subphrases were empty then we would get:

$$(\, n \quad (\,) \quad (\,) \,)$$

And if subphrase f had a single item (i.e. subphrase) then its format looks like this:

$$(\, n \, (\, (\, \ldots \,) \,) \, g \,)$$

The dabalog is a true log or rather three separate logs in one–one for dabaysms, one for the family relations, and one for other information. Once a dabarecord is introduced to the daba, its dabalog records every change to the dabarecord, and it puts a time stamp on each change.

4.7 Dabaversion 0 of the daba (official!)

The dabaversion 0 consists of only two dabarecords. The dabarecord 0 is called *everything* or even *the daba*. The dabarecord 1 is called *nothing* or *empty*. The respective dabanames are $\forall$ and $\emptyset$–they are both dabagrams. The whole daba at this time (dabaversion 0) is as follows (till the end of this section):

$$\begin{array}{l}
(\\
(\quad 0 \; (\, 2016\text{–}12\text{–}26 \; h.21{:}09 \,) \; \{ \, [\, \forall \,] \; all \; \bigvee \, \} \; (\, \{ \, \} \, \{ \, 1 \, \} \, (\, \{ \, \} \, \{ \, 1 \, \} \,) \,) \\
\quad (\, (\,) \, (\, (\, (\, 2016\text{–}12\text{–}26 \; h.21{:}37 \,) \, (\, \{ \, \} \, \{ +1 \, \} \, (\, \{ \, \} \, \{ +1 \, \} \,) \,) \,) \,) \, (\,) \,) \quad) \\
(\; 1 \; (\, 2016\text{–}12\text{–}26 \; h.21{:}37 \,) \; \{ \, [\, \emptyset \,] \; nic \; \bigwedge \, \} \; (\, \{ \, 0 \, \} \, \{ \, \} \, (\, \{ \, \} \, \{ \, 0 \, \} \,) \,) \,) \\
)
\end{array}$$

4.7.1 Parsing of dabaversion 0

Parsing should be assisted by computer software. This time let's do it by hand. The dabaversion 0 is displayed above in five lines. The opening line starts the daba with the opening left parenthesis (, and the last line consists of the closing right parenthesis). The previous line consists of the dabarecord 1. It has only the first four fields. It doesn't need the dabalog field at this time. Its daba family field features just one child, namely dabarecord 0, and that's all.

The remaining two lines, just after the opening daba parenthesis, present dabarecord 0. At th time of its creation, since it is the very initial dabarecord, it didn't have any family. Hawever, about an half hour later, the dabarecord was created too, at h.31:37, and dabarecord 1 listed dabarecord 0 as its parent. For this reason the dabarecord 0 had to record dabarecord 1 as its child, at the same time, h.21:37. Symbol + in +1 has indicated that the child (namely dabarecord 1) was added (if this child were removed then it'd be recorded as -1 (symbol minus would be used)–but that's not what has happened).

5 A definition of the dabatree

Potentially, each dabarecord n associates to itself it's own daba called n-daba. The structure of the n-daba is the same as of the daba except that the n-dabarecords have historical n-ids which are the consecutive digital strings $n.0$ $n.1$ n $n.2$ ....

Then every n-dabarecord $n.p$ may associate to itself the next level daba, namely $n.p$-daba. Its $n.p$-dabarecords have consecutive historical $n.p$-ids: $n.p.0$ $n.p.1$ $n.p.2$ ....

A dabarecord of a daba of the dabatree may name family memebrs only from its own daba, and never from any other daba. Nevertheless the historical ids from all dabas of the dabatree should be always refernced in full, e.g. as 4000.20.42, and never replaced by an abbreviated form, e.g. in the given example id 20.42 nor 42 should not be accepted by any daba of the dabatree. For instance, id 20.42 must point to the dabarecord 20.42 of the 20-daba, and it cannot be nothing else.