The MetaNumber Standard

Joseph E. Johnson, PhD

I. Overview:

1. **Critical Problem:** Digital numerical data is currently published with NO STANDARD for linking the associated units of measurement, level of accuracy, and exact meaning to the value itself. Such information is scattered in the title, row and column headings, footnotes, and often buried in text that frustrates electronic processing. Thus all digital data must be preprocessed by humans every time a value or data table is read in order to make sure that the units, error, and exact meaning are compatible with the processing programs. The necessity then for human preprocessing totally frustrates both standard daily calculations as well as artificial intelligence algorithms because the units, error and meaning are as important as the numerical value itself. With the emergence of the Internet of Things (IoT) where sensors, satellites, instruments, and humans create exponentially more massive quantities of numerical measurements, such a standard is of the greatest immediate importance for all of scientific, engineering, business, medical, and social advancement.
2. **Proposed Solution:**
    a. Establish a standard such that all four entities are linked into a single expression that can be easily read by both humans and by machines.
    b. Express all digital (and printed!) numerical data in this standard form upon its first publication, in tables, books, and all formats, thus making the full meaning of the value easily readable without error, instantly, by both machines and humans.
    c. Output all subsequent digital processing of numerical values into such a standard form.
    d. The design must give a “unique name” to every single numerical value in this standard, where this name can be used as a variable name for that value in all subsequent mathematical and logical processing. It must also allow for unlimited attached metadata without any associated processing or data transfer overhead.
    e. The processing algorithms should be optimal in terms of the highest speed of processing and the minimum storage requirement.
3. **Overview of the MN Standard**: A MetaNumber (MN) object consists of four components: (a) a numerical value (3.45e6 , 12) , (b) its units of measurement (ft/sec) , (c) a denotation of its accuracy (exact or significant to the number of digits shown), and (d) an optional metadata tag. It will also have (e) a unique reference “name” that can be used to retrieve and process that value achieved by using the internet path to the value as described below and allowing unlimited associated defining and qualifying metadata.
    a. Example: “ 3.4e9*m/s*{velocity of object A} ”
    b. Units are not present if the value is dimensionless.
    c. The accuracy (numerical uncertainty) is to be taken as the number of digits shown UNLESS there is a special indicator of an upper case “E” for scientific notation. The following are exact values: 13E0, 1.3E5 although another simple flag could indicate exact values. This notation is suggested as it is also naturally processed by modern languages such as Python. The open source module “Python Uncertainty” is included in the code and will automatically manage all error analysis among values. Note that the output of an exact value must have the default lower case “e” changed to an upper case “E” such as “309E0” or 3.456E7 adjoined to indicate that the result is exact for future calculations.
    d. The optional metadata tag (in the format {…anything …} but is optimally in the format: {var1 = val1|var2 = val2)}. The program looks for the form “{….} and converts it to a function o(…) which always returns the value “1” so when multiplying the MN it disappears and thus is just a container for metadata describing that value especially with descriptors like Lon., Lat., Depth, DateTime etc.

4. **Overview of the MN Infrastructure**:
   a. It is accessed via the link www.metanumber.com which reroutes one to the USC computer where one must register with their internet email address as the user name and a selected password. Upon submission, the secure registration system verifies the email address and then allows one to login to MN.
   b. A MN expression is a set of MN objects in mathematical expressions with indicated +, -, *, / operations delimitated by parenthesis ( ) in a valid mathematical format. MM expressions can then be submitted for evaluation (simplification to a single MN standard value) and the result is then displayed with a sequential number for that user which can be later retrieved with the expression **[my_1234]** which can be used in future expressions. All dimensional analysis, automatic unit conversions, and error analysis (via the Python Uncertainty module) are executed with error indicators for invalid unit combinations (3*ft + 2*gallon gives an error flag).
   c. Expressions can also be sent to the MN server for evaluation using API calls from ones existing software.
   d. The MN program contains names for about 700 units of measurement, dimensionless prefixes, and fundamental constants as described below along with dimensional prefix names all of which can be combined in any valid mathematical order. Thus (2.34*thousand*acre*inch + 55.67*dozen*gallon) will return the value in default metric of m3 which denotes cubic meters.
   e. User Information: Extensive descriptions and examples of usage, table of unit names ordered in different ways, and general instructional materials are available under the "Resources" tab at the top of the MetaNumber page. That tab takes one to our www.asg.sc.edu web site where one can likewise go and there access the MetaNumber tab to get the same information.

II. MN Design Details

1. **Numerical Accuracy (Uncertainty):**
   a. All values which are keyed into the MN command line are treated as accurate to the number of digits shown UNLESS the value is expressed in scientific notation with an upper case "E". Thus 2.6400, 883000, 4.1e-5, are treated as having 5, 6, and 2 significant digits of accuracy respectively, and are converted by the Python Uncertainty module to what are called 'ufloat' objects. Numbers that are to be treated as exact (to the extent that processing allows) must be indicated in scientific notation with an upper case E such as 385E0, 5.43E7, etc. Thus $130 is to be expressed as 130E0*d where 'd' is the unit of the U.S. dollar and likewise for $45.67 which must be expressed as 45.67E0*d or 4567E-2*d.
   b. This standard is not yet (Nov 1, 2016) deployed in the current software version where values are currently treated as uncertain if there is a decimal present and as exact if there is no decimal. Thus to be exact 0.0036 must currently be written in scientific notation as 36e-4 for otherwise it will be treated as an uncertain value due to the presence of the decimal. This can cause a number of problems however and we believe that a notation using the upper case "E" as described above will be the simplest method of indication. It should be noted that almost all scientific values are approximate and accurate to the number of digits shown.
   c. We are fortunate that the open source Python Uncertainty package has been developed and can simply be incorporated in our code. It also has extensions and other features that can be utilized in future MN releases.

2. **<u>Units of Measurement:</u>**
    a. <u>The MN system is based upon the SI or metric system</u> as a default and standard for definitions. These units are length: **meter** or **m**; mass: **kg** or **kilogram**; time: **s** or **sec** or **second**; electrical current: **ampere** or **amp** or **a**; temperature: **k** or **kelvin**; and luminosity: **cd** or **candela**.
    b. <u>To this standard system we have added six OPTIONAL units as follows</u>: unit of information: **bit** or **b** (a 1/0 state); unit of digital processing (indicating a double precision floating point operation): **flop** or **op**; unit of value in terms of the U.S. dollar: **d**, **usd**, **dollar**, a living human (as in 'per captia'): **p** or **person**; baryon number (e.g. number of protons or neutrons): **bn**; and lepton number (e.g. number of electrons, muons, neutrinos ....) as **ln**.
    c. <u>Units of measurement are written as variables</u> that adhere to modern standards for variables in computer science and are to have NO INFORMATION expressed in terms of the font and thus are:
        i. Always lower case (kelvin NOT Kelvin), no superscripts or subscripts (m2 for square meters and s_1 for 1/second), in the singular case (no plurals), and no Greek or special symbols (thus **hb** and not ħ and **pi** NOT π) .
        ii. Note that in the current version we have hardcoded many of the powers of some units into the system thus m2, m3, m_2, s2, s_3, ft2, ft3, etc can all be used. But this is not the optimal way to express powers. Rather in the next version we will treat any integer that follows a unit name as representing the unit to that power (or if proceeded by an underscore, e.g. _2, then an inverse power). One could then use mile3 for a volume of a cubic mile etc. without having to hardcode all powers.
        iii. Dimensionless prefixes are extensive for standard English names as well as all metric prefixes. Thus one may write expressions such as "3.4*dozen*nano*giga*exa*billion*quadrillion" as a valid expression.

3. **<u>MN Archived Data: Stored MN Values in Tables or Past Results:</u>**
    a. We reserve the brackets "[" and "]" to easily name MN values stored on the default server in tables of different dimensionalities by using the format [table name_row name_column name_...] to retrieve the value. Thus [e_gold_thermal conductivity] will retrieve the thermal conductivity of gold from the elements table named "e" and this name can be used in any mathematical expression. The mass of an elephant can be retrieved with [mass_elephant] and an electron mass with [mass_electron]. Tables naturally must have a unique name on the server in the default path. The MN standardized table is constructed to have the first row consisting of unique strings labeling each column and the first column must have unique strings that label each row. By unique names we mean strings of symbols which when case is lowered and when white space is removed, the table name and the reference label will match uniquely. To do this we created a simple function that acts on a string and removes all white space and lowers the case. Thus one can equally write [e_gold_thermal conductivity] or in camel font with some spaces and upper cases as [e_GoLD_ Ther mal Conductivity] and the retrieval will work.
    b. A table can have metadata (non-MN values) in any row or column by so indicting it with a "%" sign in front of the row or column name. In the element table we might have the name of the person who discovered the element in some column with the column name "% Discoverer". If the metadata name is unique and could be used potentially later (not in the current version) to reference a row or column then that row or column might have the form "%%Symbol" which could label the unique symbol for an element name allowing one to use the reference [e_fe_density] rather than [e_iron_density] as indicated by the "%%"name prefix.

c. The table element in the first row and first column is of the form: %%table name|table # where tables also have a unique integer number. This allows table values to also be retrieved using the format [12_44_39] to access table 12, row 44, column 39 etc for higher dimensions.

d. Since numerical data is stored in diverse directories on computers anywhere and not just the standard MN server, how can all the numbers in the universe be denoted and referenced in this manner? That is to be done by using the internet path to that value as: [internet path to the server of interest _ directory path to the directory where MN tables are stored __ table name_row name_column name]. Note that there is a double underscore that precedes the table name indicting that the previous structure is to give the internet and dir path to the file.

e. As mentioned above, all of a user's past results are assigned a sequential value which only that person can retrieve with the expression [my_356] (or the short hand of [_356] to obtain my result number 356. The system is actually keeping more extensive information with each operation by output to a file with the following values: userid, user seq #, datetime, MN input string expression, MN output string, subject. We need to have a way to output the records in this database, however this is not currently available.

f. Currently there are about 250 MN standardized tables including extensive tables from the World Bank data.

4. **Special Functions and Features:**

a. Temperature conversions cause a special problem because the offset of zero is different for the Kelvin, Celsius, and Fahrenheit scales. This offset does not allow one to follow the same algorithm that was used for other unit conversions. The Kelvin scale is the SI metric and defacto standard because it has no negative value which is essential in working with thermodynamic problems. Yet we do not want to always express the temperature in kelvin as this is not intuitive and one requirement is that the data be readable to both humans and computers.

b. We solved this problem with the functions dc( ) and df( ) which convert their arguments to kelvin automatically but which can be read by humans as Celsius and Fahrenheit respectively in the dc() or df() formats. Other conversion functions allow conversions among all temperature scales but if a temperature is to be represented in C or F then it needs to be output in the format dc() or df() in order to be valid.

c. Often one can express a large set of values as a mathematical expression which can replace a large data table. The speed of sound at 0 degrees Celsius is 331.45*m/s but the values at all other temperatures can be expressed by the expression vsound = mach = (331.45 + 0.6*tc)*m*s_1 where tc is the temperature in degrees Celsius. The use of expressions to replace large data sets can be very useful.

III. Conclusions & Vision

1. **MetaNumber Features & Advantages**:

a. The MN standard enables the fully automated retrieval, data exchange, and the reading of scientific, engineering, business, financial, commercial, and medical information for both humans and digital systems including RF tags and barcodes. (Reading a product barcode can instantly give the weight, volume/dimensions, calories, etc. in any desired units with associated metadata.)

b. This will also allow digital systems to act instantly on standardized information and incorporate such information directly into the users software with automated dimensional (unit) and error analysis with the tracking of the exact meaning of the data. (The use of API calls among software agents and apps can empower such software for processing enabling radically new designs for such software modules with automatic accuracy computations).

c. The MetaNumber standard totally solves the problem of conversion to metric and even allows users to use units of measurement that are natural to their field – even specifically designed for their business or personal use. (In every domain, units that are intuitive can be used such as solar mass, speed of light or sound, scientific constants, and natural energy units like calories, barrels of oil, kwhr, & therms – even truck & container sizes for shipping.)

2. **Additional Features for AI & IoT:**
    a. A new foundation for Artificial Intelligence (AI), Big Data, and the Internet of Things (IoT) as software can processes numerical information without human preprocessing due to the automatic conversion of units, computation of meaningful accuracy, and verification of exact meaning with traceability of processing. (Entire new levels of AI can now be written that access MN tables and process increasingly vast information without human intervention as well as supporting AI human interfaces such as Watson and Seri.)
    b. New advances are now possible with our recently designed methodology for converting standardized MN tables into two networks – one among the entities (the rows) of a data table and another the properties of those entities (the columns). Then with our additional methodology for identifying clusters in networks, our software can automatically perform our innovative cluster analysis on all MN standardized numerical information. (These clusters are automatically ranked by numerical "degree of clustering" and the dominant components of each cluster are automatically identified by defining metadata in the table. This methodology is a novel type of AI that can discover hidden order and information in numerical data tables) .
    c. Then given the capability of fully automated cluster discovery on MN standardized tables, and the associated creation of two novel networks from each table, one can then link table clusters together from different tables using the defining metadata that is attached to each cluster. (This would enable the creation of a vast universal network that links all clusters from all numerical tables, given some parameter of minimal proximity. Thus as numerical information was accumulated in all areas of knowledge, the network among the hidden information revealed by these clusters would be linked into what we envision as the "universal network of all numerical information".
3. **New software algorithms are currently being designed:**
    a. A new algorithm is being implemented that can classify the computations of users (in a group or company that so allows) by a novel analysis of the dimensional construction of the expressions submitted along with the associated units and scientific constants, MN tables retrieved, and metadata incorporated. This algorithm creates a computational profile of the equations and components used in the users R&D. These profiles can be shared allowing extensive analysis of the methodologies and work in progress within a large company or research group. (Just as the cluster linking network designed above links hidden information in the numerical data tables, this new algorithm can build networks among the underlying mathematical computations of users and identify hierarchies of investigations in say nonlinear plasma flow or novel material research in superconductivity.)
4. **Technical Infrastructure:**
    a. The MN program runs on one of two rack mounted servers in the secure USC UTS computer room named "tyson" and purchased under the ASPIRE II funding. The server has 20 GB internet access and 20 GB access to a shared 100 TB external storage. It is maintained by the USC CyberInfrastructure Group directed by Dr. Phil Moore.