Handling Various Units and Unit Representations Across Multiple Data Sources

Mihran Simonian - 12386294
mihran.simonian@gmail.com
University of Amsterdam

ABSTRACT

While the manufacturing industry attempts to implement open data communication over the whole supply chain, existing computer (ERP) systems are maintained in-place at individual manufacturing plants. The wide range of systems, together with the wide range of sensors used to measure manufacturing parameters, create multiple challenges. As supply chains operate and procure more cross-boarder than ever before, there is a growing need for a standardized units of measurements and other key parameters (such as ordering quantities). However, computer systems are hardly designed to cope with units, let alone when introducing multiple units. In this paper I propose a system to unify these units and will create a setting in which I will test the proposed system.

1 INTRODUCTION

The manufacturing industry is currently going through a major development, described as 'Smart Industry 4.0'[6]. By using computers, big data and the implementation of various sensors, manufacturers attempt to optimize their production flows and lower costs, whilst increasing reliability and optimize production speeds.

Simultaneously OEM producers (manufacturers whom combine loose parts to assemble one product) attempt to optimize their supply chain, by combining the data streams of various suppliers attempt to optimize it's supply chain, in order to avoid production halts when a disruption occurs in the supply chain, as production halts are very costly and can create significant delivery delays to the end-users.

As Haverkort stated[5];

> In many countries around the globe, research and innovation programs are starting that address this field, thereby addressing a wide variety of issues, including improved supply-chain management techniques, utilizing data acquired locally and globally.

Global Organization

Due to the fact that production facilities can possibly be spread all over the world[1], there is an increasing issue that is rising but currently less highlighted as computer systems are currently not necessarily designed to cope with multiple measurement units[3].

Physical Units

Differences in measurement units used in manufacturing industry are visible on all type of fields, yet temperatures, forces, distances, volume and weight are the main types (we can derive most others from these types). To illustrate the magnitude of the issue; in 1999 NASA lost one of it's satellites orbiting the planet Mars, due to a programming error in which a US Customary unit was implemented but the Metric unit had to be used instead[2][1]. Surely manufacturing a car or assembling parts for a lithography machine[3] will not result in the same financial (and social) consequences, but it is an indication that we should not discard the issue and it's potential consequences.

Other Units Used in Supply Chain

As we are using the case of the supply chains of manufacturers, we should not disregard the existence of other units than technical or physical units, which are important but more from a engineering perspective. Supply chain planning revolves mainly around allocating resources efficiently. Various researches [4][7] have focused on issues such as:

- Work shifts with various hourly rates
- Discount policies
- Production times
- Ordering quantities

The main point to take from this is that products or units can be reflected differently in various systems; when we order 'a box of screws Z' from company A instead of company B, the actual quantity included inside the box can differ. An important point to address when integrating supply chains, as ordering items is one of the key tasks in supply chain management.

Different Representation of Units Across Variables. ERP systems can be a mess when looking at how the same unit is represented for different variables. From personal experience I know that the ERP system which my employer utilized

[1] https://www.asml.com/en/company/sustainability/responsible-supply-chain

[2] https://mars.jpl.nasa.gov/msp98/news/mco991110.html

[3] https://www.asml.com/en/technology

(Baan IV[4]) contained different representations of the same unit for different parameters, such as with days and (again) package units. As an example; delivery times would be represented in calendar days whilst production times would be represented in work days. When integrating the various ERP systems, such differences cannot be ignored and have to be unified as well.

2 RELATED WORK

The issue of using multiple sensors and vastly storing these in computer systems has already been highlighted in previous work [8], in which data variable types were highlighted as possible sources of conflict. This was in 1990, however in 2013 there was still reason enough for concern, as [3] described in his paper.

Co-currently the Dutch research institute TNO is currently developing (test version live since January 2020) a communication protocol (labelled as SCSN) in cooperation with various manufacturers[5][6]. This communication standard enables manufacturers to share various production parameters (data) in an easy manner, whilst the individual manufactures remain to be able to use their own ERP[7] system, which is preferred by manufacturers due to various reasons (legacy purposes, proprietary software, vendor lock-ins, etc.). The SCSN protocol is introduced as it assures that companies do not need to 'give up' their own dedicated ERP systems, whilst being enabled to share data with other manufacturers (cross ERP). Thus, systems might communicate via live data connections or by means of backlogs, where on a daily night backlogs are loaded into systems (by using csv file containers for instance).

Interviews conducted with the research department of TNO revealed that the main issue currently faced for the SCSN network is getting manufacturers to trust the system and scaling up of the system. A strong point of attention remains the correct implementation of units inside the data sharing facility as not only the units themselves can differ, but the way that the same unit is used in a system can also differ.

Libraries

Unit Conversion. Unit conversion is not a new thing, even the basic Windows calculator is able to convert units. Multiple libraries for Python have been written, such as PintPy[8] or the 'unit-converter' from PyPi[9]. These libraries allow users to convert units, but only when the actual number is accompanied by a certain subset of information; the information needs to be supplied in a certain way. Thus this means, we need pre-processing steps. Not only does this create an unnecessary influx of data to be transmitted between nodes, but it also requires additional computation time to be used, burdening the system potentially unnecessary.

Other used units. To lessor extent Workdays are also a known issue in computer management. Excel offers the function 'workday'[10], for example. For Python there is a library for 'workdays'[11] as well, just as the package 'businesshours'[12]. These libraries are necessary when we want to calculate the 'earliest shipping time' as commonly used in the field of supply chain.

Complexity

The essence of unit conversion is actually not a hard concept; you take a number and then convert it according to the formula which is a known static variable. The issue is that there are many units, and they are used between each other within ERP systems. When automatically submitting an order from company A to company B, this might lead to all sorts of issues. It becomes apparent that this is (one of the main) reason(s) why various initiatives (such as SCSN) are struggling with up-scaling their potential market, as companies are not fully in trust of such a system (yet).

Current Gap in the Market

The currently available methods allow for either standard units to be converted, e.g.; kilograms to grams, Fahrenheit to Celsius and so on. These are physical units and are important, but as mentioned previously, these are not the only units that are being used within manufacturing and supply chain management.

The 'workday' and 'businesshours' libraries as currently available on their part do not allow to convert units.

There is no research conducted yet in which these two important items can be combined into one, by applying both packages on a single tabulation, thus providing an incremental improvement for the overall research of combining the various ERP systems.

3 RESEARCH

The framework of this research is having multiple ERP systems combined, as the SCSN network is attempting to achieve.

[4] https://www.xibis.nl/producten/baanivenv

[5] https://smartindustry.nl/fieldlabs/8-smart-connected-supplier-network

[6] https://www.brainportindustries.com/nl/berichten/maakindustrie-aan-de-slag-met-digitalisering

[7] https://en.wikipedia.org/wiki/Enterprise_resource_planning

[8] https://pint.readthedocs.io/en/0.11

[9] https://pypi.org/project/unit-converter

[10] https://support.office.com/en-us/article/workday-function-f764a5b7-05fc-4494-9486-60d494efbf33

[11] https://pypi.org/project/workdays

[12] https://pypi.org/project/BusinessHours

Aim

This research is aimed to highlight the importance of identifying the need to convert a wide range of varied units, represented in various data sources. By performing a performance test we will be evaluating the impact of combining various units into a single large tabulation.

Relation to Big Data

The angle of approach for the current paper is designed around the 'variety' aspect of the Big Data set of V's. The question how to reduce variety of units, which can potentially lead to all sorts of practical problems (losing a satellite in space) is key and as a result we will also notice what impact all these various units (together with the usage of the library packages) will have on translation speeds. This leans a bit towards the V of 'velocity', a logical consequence as more variety leads to less available computation power. The main focus however remains to be on 'variety'.

Research Question

During this research I would like to answer the following question:

- How do we combine various units and translate them into a single representative unit using Python libraries?

The following sub questions will also become part of the research:

- Which type of data variables do we want to align in the current scope?
- Which possible inputs can we expect?
- Which problems might arise during the scale-up of the system?

4 METHOD

Data Sources

For this research I will use a self-generated dataset upon which the introduced method will be tested. The dataset is generated using Microsoft Excel as it enables us to generate random numbers rather quickly, and convert them to a set of datatypes. Also Excel allows us to export data to a variety of datatypes.

- csv
- txt

Dataset and Personal Experience

The dataset will be generated based upon personal experience from working with an ERP system which used the ASML products. The dataset therefore contains the ASML product part number system labelled '12NC'[2]. This identifier is unique per each specific product, from small items such as rings and screws to large items like complete machines. This provides us with a unique feature to interlink the items within the dataset, which comes in hand when we want to verify whether the implemented method actually works.

Unit Types

For the current research I will solely focus on the application of the following units:

- Temperatures
- Mass
- Earliest Shipping Day

Python

For the implementation the Python language is used as it allows us to work with various libraries as described under the relevant work section. Python also allows us to work with Jupyter Notebooks[13], a interactive interpreter which is very suitable to Python.

Libraries. The following Python libraries will be used:

- Pandas
- Numpy

The Method

The method to be designed will be using a combinations of libraries, though it is unclear at the moment which libraries will specifically be used. The pool of selection is:

- PintPy
- PyPi: Unit-converter
- PyPi: Workdays
- PyPi: BusinessHours

Scope and Limitations

This research is a mere introduction to the vibrant world of (measurement) units and is intended to highlight the importance of unit alignment. There are for instance differences between UK (Imperial) and US (Customary) units[14], even though they use very similar notations (to add to the confusion). I will however not dive very deep into this topic as it does not add much to the model itself, it's a mere iteration of an existing situation.

Furthermore this research will not try to optimize and reduce the computational cycles as required in order to transform units, as this would transform te research more into reviewing this unit issue from the big data perspective of 'velocity' oppose to the intended 'variety'.

[13] https://jupyter.org/

[14] https://en.wikipedia.org/wiki/Comparison_of_the_imperial_and_US_customary_measurement_systems

ACKNOWLEDGMENTS

To Hannes Mühlheisen, for the fun and joy during the lectures and for setting up this exercise, together with Cristian Rodriguez Rivero and Shuo Chen.

REFERENCES

[1] MCO Mishap Investigation Board. 1999. *Mars Climate Orbiter Mishap Investigation Board Phase.* https://llis.nasa.gov/llis_lib/pdf/1009464main1_0641-mr.pdf

[2] N Niki de Kadt, TJG Peeters, F Langerak, and AA Alblas. 2015. Accelerating the Learning Curve at ASML.

[3] Marcus P Foster. 2013. Quantities, units and computing. *Computer Standards & Interfaces* 35, 5 (2013), 529–535.

[4] Stephen C Graves, David B Kletter, and William B Hetzel. 1998. A dynamic model for requirements planning with application to supply chain optimization. *Operations Research* 46, 3-supplement-3 (1998), S35–S49.

[5] Boudewijn R Haverkort and Armin Zimmermann. 2017. Smart industry: How ICT will change the game! *IEEE internet computing* 21, 1 (2017), 8–10.

[6] Jay Lee, Hung-An Kao, Shanhu Yang, et al. 2014. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp* 16, 1 (2014), 3–8.

[7] Biswajit Sarkar. 2016. Supply chain coordination with variable backorder, inspections, and discount policy for fixed lifetime products. *Mathematical Problems in Engineering* 2016 (2016).

[8] Edward Waltz, James Llinas, et al. 1990. *Multisensor data fusion.* Vol. 685. Artech house Boston.