Gist issue 451: simplify Magnitude and Unit of Measure

Phil Blackwood

May 2021

Table of Contents

[Motivation and approach 1](#_Toc72740514)

[Summary of improvements 2](#_Toc72740515)

[Summary of the proposal 3](#_Toc72740516)

[Main usage patterns 6](#_Toc72740517)

[Doing math with magnitudes 9](#_Toc72740518)

[Standard Units 10](#_Toc72740519)

[Example of adding a new unit of measure 14](#_Toc72740520)

[Comparison with gist 9.6.0 16](#_Toc72740521)

# Motivation and approach

Gist is the de facto Enterprise ontology, but not yet widely adopted. A simpler version of gist will lower barriers to adoption and help prevent mistakes on the part of users. The objective of the simpler version of Magnitude and Unit of Measure presented here is to make a portion of gist “as simple as possible, but not simpler”.

This simpler version of gist Magnitude and Unit of Measure originated from training exercises and the process of learning gist 9.6.0. A student of gist (like myself) can find the meaning of the following terms difficult to remember and apply correctly:

has StandardUnit (whose value is a coherent unit)

convertToStandard (used to convert to a coherent unit)

simple unit

base unit

product unit

ratio unit

coherent product unit

coherent ratio unit

multiplier

multiplicand

etc.

The terms standard, base, and simple, for example, can tend to blur together between visits to the definitions. Parsing complex units using multiplication and division can also be somewhat awkward and turns out to be problematic for other reasons.

In search of a simpler model, the approach taken to was to understand gist 9.6.0, read online literature about the International System of Units and the QUDT ontology (Quantities, Units, Dimensions, and Types originating with NASA), and then from the ground up try to construct the simplest possible representation of Magnitudes and Units of Measure that would work.

The resulting proposal has 15 concepts instead of the 55 in gist 9.6.0. A later section of this document gives a detailed comparison.

But of course, it’s only an improvement if it works; a large part of this document goes into the detail needed to evaluate how well it meets the need.

# Summary of improvements

The bottoms-up analysis has identified what appear to be latent issues with gist 9.6.0. If these are framed as requirements, the list is:

1. support a full range of units, including business metrics, business KPIs, units used to define and manage products and services, and units for science and engineering (a subset is included in the International System of Units).
2. address the issue raised in the International System of Units: “several different quantities may share the same SI unit” and similar cases such as mean-time-to-repair and mean-time-between-failures share the same SI unit. Another example is width, height, and depth of kitchen cabinets share the same SI unit.
3. do not require new classes to be created for each business metric. Do not require any new classes to be created, unless a user finds it useful for maintaining a taxonomy of aspects (or possibly units).
4. allow the numerical value of a magnitude to be fractional, e.g. 1/3 of an acre
5. allow conversion factors to be fractional
6. if decomposition to base units is needed, support fractional exponents (fluid dynamics)
7. provide simple rules for determining what calculations may be done on a collection of magnitudes
8. provide a standard way to represent average and standard deviation
9. allow a magnitude to refer to an external taxonomy to describe what type of measurement it includes
10. make sure fractions and percentages can be accompanied by information about “what is a fraction of what” or “what is a percentage of what”
11. support scales that involve scores like net promoter score
12. align with the latest version of the International System of Units
13. improve annotations

There may be many ways to support these requirements. In the following proposal, the way they are supported is as follows:

* use the Aspect class to identify what characteristic is being measured: supports 1, 2, 3, 8, 9
* introduce numericalValue to allow fractional values (new property proposed by Michael Uschold): supports 4
* define standard unit as having a distinct physical meaning: supports 2
* if decomposition to base units is needed, do it as a product of powers: 6
* expand the range of conversionFactor: supports 5
* remove the class for base units: supports 12
* Rename “each” to “one”: supports 12
* Fix description of Magnitude: supports 13
* Fix example of conversion from Fahrenheit to Kelvin (one place says to add the offset, and then the example shows it being subtracted): supports 13
* Document “math rules” either in scopeNotes or supplementary documentation: supports 7, 13
* tbd: 10

# Summary of the proposal

The concepts are as follows:

*Magnitude* is a simple concept that is easier to use than it is to explain. In each example below, the magnitude is underlined:

“The *size of lot #2176* is 1/3 of an acre.”

“The *size of the audience for the show* was estimated as 3 million viewers.”

“*Our current average cost to acquire a customer* is $300.”

“The *width of cabinet model b2724* is 27 inches.”

A magnitude is described by four properties:

*categorizedBy* some Aspect

*hasUnitOfMeasure* some UnitOfMeasure

*numericalValue* some number (typically a double like 5.68, or a ratio like 1/3)

*hasPrecision* some Magnitude (allows the estimated accuracy of a measurement to be recorded)

A magnitude can involve a single measurement, an average, or a reference value like a speed limit.

*Aspect* (a measurable characteristic) is also easy to use; below are some examples:

*Area* can be measured in acres.

*Number of viewers* can be found by counting or estimating.

*Cost to acquire a customer* can be measured in dollars.

*Width* can be measured in inches.

Any business metric is an aspect, typically measured or specified as a time duration, monetary value, or count.

Engineering and scientific domains use aspects like *electrical current* or *specific heat capacity*.

The values of Aspect will reflect the terminology of a domain, which can vary considerably. The simple concept of distance, for example, could manifest itself in aspects like perimeter, elevation, altitude, depth of channel, orbital perigee, etc.

[ Per Dave McComb, Aspect was originally intended for “folksonomy” type characteristics such as cabinet width. In the proposal, there is intentionally no set of “standard” aspects, following a “when in doubt, leave it out” approach. The expectation is that users know the meaning of the units and what they refer to, for example an electrician understands what an amp is. ]

*UnitOfMeasure* is what it sounds like:

Area can be measured in *acres*.

Number of viewers can be found by counting or estimating (the unit is *one*).

Cost to acquire a customer can be measured in *US dollars*.

Width can be measured in *inches*.

A unit of measure can be further described with the properties:

*symbol*

*symbolHTML*

*symbolUnicode*

*hasStandardUnit* some StandardUnit

*conversionOffset*

*conversionFactor*

*StandardUnit* is a subclass of UnitOfMeasure based on the International System of Units, but refined to make it easier to avoid “comparing apples and oranges”.

*ampere*, *candela*, *kelvin*, *kilogram*, *meter*, *mole*, and *second* are standard units from the International System of Units. Each of these has a corresponding aspect (electrical current, luminous intensity, etc.)

*one* is a standard unit as described in the International System of Units.

*bit* is a standard unit added by gist to measure an amount of data.

*US Dollar* is a standard unit added by gist to measure monetary value. Unlike the other standard units, US Dollar has no physical meaning.

This proposal defines a *standard unit* as any of the units above, or a unit that is equivalent to a product of powers of units listed above that has a distinct physical meaning. There are about 75 standard units that can be identified using the International System of Units, including the “special named derived coherent units”. Examples:

*ohm* to measure electrical resistance

*volt* to measure electrical potential (usually called voltage)

*Joule per kelvin of entropy*

*Joule per kelvin of heat capacity*

*gray of absorbed dose*

*gray of kinetic energy released per unit mass (kerma)*

*sievert of dose equivalent*

Note that the last three are all measured in square meter per square second, but are treated as separate units because they measure physically different aspects. This approach is extended to other cases where a unit has multiple physical meanings.

Electricians work with amps, watts, volts, and ohms. To use these standard units, they do not usually need to know the equivalent in kilograms, meters, and seconds. It’s really not a problem if your electrician does not know what 1 kg m2 s-3 A-1 is.

Decomposing standard units to their equivalent expression in base units is analogous to factoring a number into prime numbers. It helps define the standard units, but is not necessary to use them, just we can use the number 1492 without knowing its factorization, or measure volts without understanding its breakdown into base units.

Relying too much on the equivalent expression in base units can be hazardous to public health; it is the reason the International System of Units created separate names for becquerel, gray, and sievert.

*conversionOffset* and *conversionFactor* are used to convert non-standard units like inches or miles to standard units like meters.

Conversion rule: add the conversionOffset (default zero), then multiply by the conversionFactor.

To convert from standard to non-standard, reverse the steps: divide by the conversionFactor, then subtract the conversionOffset.

# Main usage patterns

In general, simpler units are associated with a larger number of context-specific aspects. Business metrics are at the simpler end of the spectrum, and engineering metrics extend into the more complex end.

Consistency between unit and aspect can be established by relating aspects to standard units (e.g. with the *hasStandardUnit* property). In the diagrams below, the aspects on the left side are consistent with the units on the right side because they point to the same standard unit:



As they define aspects, Enterprises can use their existing terminology.

Business metrics tend to have simple units of time duration, monetary value, and pure-numeric values like counts, ratios, and percentages. They can include averages over a collection, and reference values such as business targets. In general, dozens of standard business metrics will have the same unit of measure.

A business typically has products and services that are described with magnitudes. The ones with simpler units may have more corresponding aspects, while the ones with more complex units tend to have fewer, more standardized aspects.

Ratios and percentages should include the aspect of numerator and denominator to supply meaning. (how to do this is tbd)

An Enterprise can profile its existing data to start to normalize units of measure.



The profiling should extract aspects as well as units.

Gist could have a pre-defined set of units; the user could select simple units they want (inch, foot, mile) and the aspects they want (radiant flux, ACV, etc.) and the list of all units with conversions could be populated into their knowledge graph. Or units could be set up automatically during data profiling.

Statistical metrics can be calculated whenever the units are the same. We can think of an average as an aspect of a collection and use it in combination with any other aspect or unit of measure. For example:



It is common to want to know how accurate measurements are. We can think of *precision* as an aspect of a measurement method and, like an average, we can use it in combination with various aspects and units of measure. For example:



For many practical applications, the errors in measurement follow a normal curve. You might want to adopt a convention that precision represents 2 standard deviations of the error distribution. Then, for a well-calibrated method, there is about a 95% probability that the error in your measurement (absolute value) is less than the precision:



The precision is commonly stated as a “plus or minus” value, e.g. weight is 519 kg plus or minus 0.5 kg, meaning there is a 95% chance the actual value falls within that “plus or minus” range.

If a client needs to do dimensional analysis, properties for *factor* and *exponent* could be introduced to support expressions in terms of base units or aspects (F = ma, for example).

# Doing math with magnitudes

An Enterprise will want to use quantitative data in calculations and comparisons to answer all kinds of questions. In this section we look at ways to help ensure that the calculation or comparison makes sense.

Quantities used in multiplication and division are not required to have the same unit of measure:

3 feet \* 4 pounds = 12 foot-pounds

$12 / 4 widgets = 3 dollars/widget

Addition, subtraction, averages, standard deviations, min, max, greater-than, less-than, and equal-to, only make mathematical sense if all the numbers have the same unit (or have been converted to the same standard unit):

3 feet + 4 feet = (3 + 4) feet = 7 feet

(3 feet + 4 watts) does not make mathematical sense, or physical sense

A calculation in which the units are all the same, and the aspects are all the same will usually make “business sense” or physical sense:

average cost of customer acquisition (all in the same units)

average luminous flux (all in the same units)

… but not always:

cabinet1 depth + cabinet2 depth + cabinet3 depth is probably not useful during kitchen design (but possibly for shipping).

Having different aspects can be used as a caution that a result might not make sense:

average of (costs of customer acquisitions and costs of materials) could trigger a warning

… but different aspects do not always indicate an error:

cabinet height + countertop thickness makes sense

Given this set of examples, any set of constraints to be put on calculations based on aspect should be carefully thought out. The ontology itself does not constrain the way the data is used; SHACL can be used for constraints where needed.

# Standard Units

Gist standard units are based on the International System of Units, with some enhancements. Gist adds bit and US Dollar. Other differences are described below.

Celsius is treated as a coherent unit in SI, but not included as a standard unit because it measures the same physical characteristic as Kelvin. In gist, every standard unit (except US Dollar) has a distinct physical meaning.

Velocity is treated as an aspect in SI, but this is not consistent with the definition in physics. A velocity should be thought of as a thing that has a direction and a speed, so velocity is a “thing” that has a magnitude (the speed), rather than an aspect.

The SI Brochure cautions that some units only have physical meaning when paired with their aspect:

"several different quantities may share the same SI unit"

See the [Brochure](https://www.bipm.org/en/publications/si-brochure), section 2.3.4 for more explanation.

Rather than rely on code to pair up the unit with the aspect to interpret the data, the pairing is done by defining a separate unit for each physically distinct aspect (business metrics will be discussed separately). For example:

The unit *watt* and the aspect *electrical power* are combined to form *watt of power*.

The unit *watt* and the aspect *radiant flux* are combined to form *watt of radiant flux*.

When a Magnitude is created for electrical power, *watt of power* should be selected, and this will help prevent calculations that would conflate it with a watt of radiant flux.

[ also consider interoperability, since SI is a well-established standard. Interfacing with an SI compliant application requires exchange of unit and aspect to ensure physically meaningful units. An initial evaluation seems to suggest the approach above is ok. Check to see if names like watt are normative in SI. ]

Here are the cases in which physically distinct aspects share the same SI unit, and where separate standard units should be created:

electric displacement, electric flux density, and surface charge density (A s m-2)

energy density, pressure, stress (kg m-1 s-2)

energy, amount of heat, work, moment of force (kg m-2 s-2)

entropy, heat capacity (kg m2 s-2 K-1)

molar entropy, molar heat capacity (kg m2 s-2 mol-1 K-1)

power, radiant flux (kg m2 s-3)

heat flux density, irradiance (kg s-3)

electrical conductance, capacitance, permittivity (kg-1 m-2 s3 A2)

absorbed dose, kerma , specific energy, dose equivalent (m2 s-2)

specific entropy, specific heat capacity (m2 s-2 K-1)

angular frequency, angular velocity (rad s-1)

activity referred to a radionuclide, frequency (s-1)

The resulting set of proposed standard units is:

|  |
| --- |
| ampere |
| ampere per meter |
| ampere per square meter |
| becquerel |
| bit |
| bit per second |
| candela |
| candela per square meter |
| coulomb |
| coulomb per cubic meter |
| coulomb per kilogram |
| coulomb per square meter of electric displacement |
| coulomb per square meter of electric flux density |
| coulomb per square meter of surface charge density |
| cubic meter |
| farad |
| farad per meter |
| gray of absorbed dose |
| gray of kinetic energy |
| gray per second |
| henry |
| henry per meter |
| hertz frequency |
| joule of energy |
| joule of heat |
| joule of work |
| joule per cubic meter |
| joule per kelvin of entropy |
| joule per kelvin of heat capacity |
| joule per kilogram kelvin of specific entropy |
| joule per kilogram kelvin of specific heat capacity |
| joule per kilogram of specific energy |
| joule per mole |
| joule per mole kelvin of molar entropy |
| joule per mole kelvin of molar heat capacity |
| katal |
| katal per cubic meter |
| kelvin |
| kilogram |
| kilogram per cubic meter |
| kilogram per square meter |
| lumen |
| lux |
| meter |
| meter per second |
| meter per square second |
| mole |
| mole per cubic meter |
| newton |
| newton meter |
| newton per meter |
| ohm |
| one |
| pascal of pressure |
| pascal of stress |
| pascal second |
| radian |
| radian per second of angular frequency |
| radian per second of angular velocity |
| radian per square second |
| second |
| siemens |
| sievert of dose equivalent |
| specific volume unit |
| square meter |
| steradian |
| tesla |
| US Dollar |
| volt |
| volt per meter |
| watt of power |
| watt of radiant flux |
| watt per square meter of heat flux density |
| watt per square meter of irradiance |
| watt per square meter steradian |
| watt per steradian |
| wavenumber unit |
| weber |

Units can be added to the set of standard units, as long as they have a unique decomposition into the base units and also have a distinct physical meaning.

For business metrics, the guideline is to establish a set of metrics with distinct meanings and check to see if a calculation includes a single business metric (aspect). ~~The alternative is to fold the metric into the unit, e.g. hours-of-mean-time-to-repair as a unit.~~ [this alternative is not a good idea]

# Example of adding a new unit of measure

A U.S. Enterprise dealing with automobiles needs to record information about watt-hours per mile for electric cars.

Watt-hours per mile is a unit of measure.

The IRI could be :\_watt-hour\_per\_mile, with symbol “wh/mi”.

They can compare any two values, because the units are the same, e.g. car model A has the best efficiency at 250 wh/mi (where a lower number is better), and the next best is 10 wh/mi more at 260 wh/mi.

There is initially no need to convert to standard units.

Later, they find they want to track watt-hours per kilometer and they want to make comparisons with the values in watt-hours per mile.

Although it would be easier for them to convert directly from one to the other, the ontology only supports conversion to a common standard unit based on coherent units.

A guideline for manual conversion would be as follows:

Look up list of base units (e.g. in annotation of StandardUnit or online):

* Hour is not standard. It maps to second.
* Mile is not standard. It maps to meter.

The standard unit is always formed by plugging in base units to substitute for non-standard ones, e.g.

watt-*hour*\_per\_*mile* hasStandardUnit watt-*second*\_per\_*meter*

They decide the symbol will be “watt-second per meter” (but do not plan to display it).

Conversion can then be done by substitution, using the mathematical expression they find most natural:

Watt hour / mile

(watt to watt) (hour to second) / (mile to meter) =

(1)(3600) / ( 1609.344) =

2.237

This is how manual conversion works, and it does not require the user to understand how to decompose the unit explicitly.

It looks like automation requires a pre-defined structure that some users will not understand, or some clever parsing. For example:

watt hour mile-1 will not make sense to some people

decomposition using multiplication and division can be done, but may prove to be error-prone because division is not commutative.

Requiring a user to define a product of powers or a decomposition by multiplication and division may be as much work, and possibly more error-prone, than using the guideline above.

From a user point of view, automation might ideally take the form of a centralized service that maintains a curated list of thousands of units and helps a user pick the ones they care about, and also provides a guided experience for defining new ones that then become part of the curated list.

Note about ProductUnit and RatioUnit: a single unit can end up being one or the other, so it’s not clear if these classes are helpful. For example:

(watt x hour)/mile = watt x (hour / mile)

# Comparison with gist 9.6.0

Classes and properties:



It is expected that as an Enterprise includes more and more of its aspects in the data (abox), it might want to group them into categories by creating subclasses of Aspect. While this taxonomy can vary from one Enterprise to another, it might include a natural split between business metrics and the quantities associated with products and services. An aspect can belong to more than one of the subclasses, providing a great deal of flexibility.

In comparison, in gist 9.6.0 it looks like a user would have to add two new subclasses every time a new aspect is introduced, one under Magnitude and another under Unit of Measure. The subclasses of Unit of Measure could end up with only a handful of individuals.

Use the Aspect class. This is especially important for business metrics and KPIs, where hundreds of different types of quantities are measured by only a few units of measure (e.g. cost to acquire a customer, value of a contract, net profit, etc. all measured in US Dollars).



Using Aspect reduces the number of classes a user needs to create to make sense of the data. There is no need to duplicate each “dimension” within subclasses of Magnitude and Unit of Measure (compare how the concept of area is modeled):



There is a single simple method of doing unit conversions, with simple rules, and a way to ensure that a unit is physically meaningful to prevent comparing “apples to oranges”.



If unit decompositions are needed, the method in the International System of Units should be used (factor as a product of powers), which gives a unique decomposition (modulo commutativity) and supports fractional exponents.



Perhaps a more detailed comparison is possible, but it looks like there is a choice between a simple ontology that works and a more complex one that has some shortcomings.