Using RASE semantic mark-up for Normative, Definitive and Descriptive knowledge

N. Nisbet & L. Ma
The Bartlett School of Sustainable Construction, UCL, London, United Kingdom

ABSTRACT: The RASE methodology has gained some attention as a means to expose the logical objectives and individual metrics found in regulatory and contractual documents. This paper will explore the role of RASE in exposing not only normative documents but also both definitive resources such as dictionaries, thesauri and classification tables, and descriptive resources such as BIM models, contract diaries, product data or technical journalism. A single common execution framework allows any of these resources to be cross-compared, independent of domain or language. This support for mixed modalities opens the potential for RASE to be used as a core concept across multiple domains and information types.

1 INTRODUCTION

The Architectural, Engineering and Construction (AEC) industry has been relatively slow to adopt digitalization compared to other sectors. This means that the construction sector retains a heavy dependance on documents not only as evidence such as certificates and photographs but also for holding the primary sources of information, even when there are alternatives available such as Building Information Modelling BIM and its evolution into asset information management. RASE (AEC3 2021) makes explicit the logical structure within knowledge content of documents and other stored formats. Previous work has examined how RASE semantic mark-up of documents can act as a precursor to a wide range of existing applications and knowledge representations, by treating them as presentations of RASE knowledge (Nisbet 2022).

In contrast, this paper examines how the RASE semantic mark-up can used to render the content of documents directly operable, without the need for any intermediate representations.

This should offer advantages in terms of accuracy and efficiency. It can also offer advantages in terms of privacy and security for knowledge content owned by governmental and commercial entities.

Table 1 gives a breakdown of types of content found in documents. Individual documents may contain several types of content. The content may be text or tables. Documents may also contain multi-dimensional and complex information. The types of content in italic are not considered further.

Table1: Type of document content

Breakdown	Document content
1	Descriptive (picturing)
1.1	Description
1.2	Illustration
1.3	Narrative
2	Definition
2.1	Synonyms and translations
2.2	Classification
2.2	Equations and algorithms
3	Normative (expectations)
3.1	Regulations
3.2	Requirements
3.3	Recommendations
4	*Argumentation (convincing)*
4.1	*Argument*
4.2	*Comparison and contrast*
5	*Exposition (explaining)*
5.1	*Analysis*
5.2	*Cause and effect*
6	*Evidence (verification)*
6.1	*Audio-visual content*
6.2	*Certificates and affidavits*

The current scope of interest has been set around normative, definitive and descriptive knowledge. This is to exclude argumentation and exposition (types 4 and 5) where the knowledge content is dynamic and potentially inconsistent, though any conclusion derived from an argument or exposition may be considered as knowledge and marked-up. It also excludes the evidential content of documents (type 6) which may be supportive of knowledge content but is not subject to reasoning.

2 METHOD

This paper adopts a design-science paradigm to allow the exploration of the domain. It offers working descriptions of the three kinds of explicit knowledge in order to test their utility. It presents the relevant algorithms developed in a series of experimental applications for exploiting that knowledge both in isolation (section 3) and in combination with other knowledge resources (section 4). Each experiment is reported with source material, its RASE mark-up and an outline of relevant algorithms. Limitations are noted.

3 THREE KINDS OF OPERABLE KNOWLEDGE

At least for the three kinds of knowledge in scope, RASE asserts that there are four roles performed by the metrics found in phrases and the sections found in the structure of written documentation. The metrics and sections may or may not exactly match the presentational structure of the document, so, for example, an exception section may be a separate paragraph following the main requirement.

Table 2: RASE Types used in mark-up

Sections	Metrics
Requirement Section	①Requirement (normative)
	②Reference (definitive)
	③Report (descriptive)
Application Section	Application
Selection Section	Selection
Exception Section	Exception

The addition of RASE mark-up to a document or its association to tabular and multi-dimensional data, creates a simple hierarchy. Nisbet et al (2022) showed that this hierarchy can be mapped to other representations using a simple tree traversal algorithm which ensures that each objective section and every metric test is visited methodically. When executing (as opposed to reporting) over a RASE document it may not be necessary to visit every section or metric and so further heuristics can be safely used to accelerate the processing. RASE as originally described (Nisbet 2008) was focused on normative knowledge (type 3). In applying that knowledge it has been necessary to also consider definitive knowledge (type 2) and descriptive knowledge (type 1).

Table 3. RASE knowledge constituents

Knowledge type	High levels	Lowest level
Normative	section	metric
Definitive	concept	term
Descriptive	entity	property

3.1 *Normative knowledge*

In response to issues around accuracy and efficiency as reported in automated regulation code compliance checking, Nisbet (2008) proposed the RASE methodology as a means to capture and render operable the normative content of Building Codes and Regulations, thereby eliminating the requirement for domain expertise, code expertise and model expertise to come together to re-interpret the regulations. Examples of normative content include much of the legal, regulatory and contractual documents. It also includes other requirements from clients or third parties. They are characterized by ‘Requirement’ knowledge content (table 2). A feature of the formal style of some legal and regulatory content is the complexity of the chapter, paragraph and sentence structures.

Normative knowledge content acts to create expectations. These expectations may be met by descriptive knowledge content such as a BIM model, or the knowledge of a user, or by measurements taken from the real world.

An application (AEC3 2022) can traverse a normative document to convert the text and tables into an interactive checklist. The checklist can include the original text with added input boxes to allow a user to answer the questions implied by the metric phrases. This is particularly useful if the normative knowledge applies to a single entity, such as a proposal, site or building overall. Initially the overall result is ‘unknown’. As each metric is answered, the document can update the overall result or hide sections that have been satisfied as ‘as required’, ‘excepted’ ‘not applicable’ or ‘not selected’, or ‘false’ If any result is ‘unknown’ then it and the relevant sections and metrics below remain visible. More work is required to decide how the outcome can be preserved and documented.


3C.9 Maximum building height

The maximum height for a dwelling house (development dwelling house) unknown unknown and any attached (development attached) unknown unknown development (type development) unknown unknown is 8.5m above ground level (existing) (height above existing ground <= 8.5 m) unknown unknown .

unknown

Figure 1: Example regulation as a form (NSW 1995)

Table 4: RASE mark-up used in figure 1

RASE type	Metrics
Selection	development is 'dwelling house'
Selection	development is 'attached'
Application	type is 'development'
Requirement	height above ground <= 8.5m

Table 5 shows the variables and abbreviations used in algorithms shown in tables 6, 7 , 8 and 9:

Table 5: Variables and abbreviations

Name	Description
tfu, t, f, u	one of true false or unknown
R A S E r a s e	RASE sections and metrics

RASE can be parsed by any logical engine that can evaluate any metric to true, false or unknown, as in equation 1 and evaluate any section on the basis of the sections and metrics below it as in equation 2 in table 6:

Table 6: Variables and abbreviations

Name	Table	Description
Equation 1	7	r a s e = tfu
Equation 2	8	R A S E = or(and(R… r…), notand(A… a…), notor(S… s…), or(E… e…))

The algorithm to evaluate any metric depends on the comparators that are supported but table 7 gives an example.

Table 7: Metric evaluation

tfu evaluateMetric(m1)
v = m1.value
g = m1.target
c = m1.comparator
if c is ">=" then return (v >= g)
if c is ">" then return (v > g)
if c is "<=" then return (v <= g)
if c is "<" then return (v < g)
if c is "=" then return (v == g)
if c is "!=" then return (v != g)
return u

The algorithm used to evaluate any section is also relatively simple as shown in Table 8: it can be is triggered by changes to a dependent metric or section.

Table 8: Section evaluation

tfu evaluateSection(s1)
// for each section or metric below
for each s1.smi
c = smi.raseType.firstChar
n(smi.tfu, c) ++
// if not applicable, pass
if c is A and smi.tfu is f
then return t
// if excepted, pass
if c is E and smi.tfu is t
then return t
// nothing selected so pass
if n(t,S) is 0 and n(u,S) is 0 and n(f,S) is not 0 then return t
// as required so pass
if n(f,R) is 0 and n(u,R) is 0 and n(t,R) is not 0 then return t
// no unknowns so fail
if n(u,A) + n(u,S) + n(u,E) + n(u,R) is 0 then return f
// there are significant unknowns
return u

The algorithm can be developed in to take into account other events such as the completion of the iteration over all the dependent metrics and sections of the same RASE type (Table 9). This is an example of a heuristic that can reduce the number of metrics that need be evaluated, which may be significant if the evaluation process for some metrics on some entities has high intensity.

Table 9: Sub-section evaluation

```
tfu evaluateLastRASE(raseType)
c = raseType.firstChar
if c is R
    if(n(f,R) > 0) return f
    if(n(u,R) > 0) return u
    return t
if c is A
    if(n(t,A) > 0) return t
    if(n(u,A) > 0) return u
    return f
if c is S
    if(n(f,S) > 0) return t
    if(n(u,S) > 0) return u
    return f
if c is E
    if(n(t,E) > 0) return f
    if(n(u,E) > 0) return u
    return t
return u
```

Other heuristics can be applied, for example based on prioritizing sections and metrics on the basis of past experience such as the ratio of their discrimination (their ability to be decisive) and their intensity (the duration of their computation). Such information would only be available in a production deployment.

3.2 *Definitive knowledge*

Some regulatory documents may include definitive knowledge setting up phrases or classification with specific meanings. Separate documents may also offer definitions of terms which may not necessarily be comfortable for the regulators or regulated participants. A regulation written in a human language may not have an obvious correspondence to descriptive model, especially if the model is held to a particular schema such as IFC (2018).

This implies that there may be knowledge content that is definitive in its intention, providing synonyms, language equivalents and bridging between different contexts. Some terms may be defined by reference to a remote service or algorithm.

Example of definitive knowledge content include formal definitions, dictionaries, classification tables, formulae, and extended algorithms. They are characterized by 'Reference' knowledge content (see table 2).

Declarative content can be a resource in automated code compliance checking and is central to semantic enhancement. A characteristic of declarative content is that there may be a selection of methods of triggering a particular clause and there may be may be several separate outcomes. A feature of the execution of 'Reference' metrics is that they must be evaluated last, so that they only take effect if there is no other means discovered of satisfying the clause.

An example of semantic enhancement could consider a clause to classify circulation space (table 10). The clause may start with defining its subject scope. If satisfied, then in response the following 'Reference' terms can be asserted as true predicates.

Table 10: Example of enhancement

RASE type	Metrics
Application	Entity is Space
Selection	Description is Corridor
Selection	Description is Passage
Selection	Description is Corr.
Selection	Length/Width > 4
Selection	Uniclass is SL_90_10_15
Selection	Is Corridor is True
Exception	Situation is not Internal
Exception	Door Count is less than 2
Reference	Description is Corridor
Reference	Uniclass is SL_90_10_15

These outcomes may create a supplementary descriptive knowledge resource held in memory or as a supplementary descriptive document. Beach et al (2013) used definitive content to set the maximum number of points achievable and the number of points achieved in performing an environmental assessment (BRE 2018).

There is potential for BIM models to carry full classification information, even though BIM authoring applications may make it difficult to apply this knowledge systematically to spaces, tasks or products. A characteristic of most tables is that each level in the classification hierarch has a definitive 'coding' linked to a 'description' which may contain words and phrases indicating its applicability, selection and sometimes exceptions, In all cases, any immediate child classifications are themselves exceptions. Further work could demonstrate updating a model with the values of properties implied by a classification code.

Figure 2: Example of definitive knowledge (Uniclass 2022)

Table 11: RASE mark-up used in figure 2

RASE type	Metrics
Application	Entity is Space
Application	Type is Circulation
Reference	Uniclass is SL_90_10
Exception Section	-
Application	Description is Corridor
Reference	Uniclass is SL_90_10_15

Using RASE, semantic enhancement rules can be fully definitive rather than procedural, so that a mixture of shape, relationship, classification or property applicability, selection and exceptions may trigger the enforcement of a variety of shape, classification, property or even relationship values.

3.3 *Descriptive knowledge*

Descriptive knowledge is separate from the real world but is expected to be a reflection of it. Descriptive knowledge may reflect a static view of reality, or it may include narrative knowledge taking a time or process view as found in a construction plan. The description can be of an envisaged future state, such as a proposed building.

Examples of descriptive knowledge content include reports, stories, and many kinds of models and data. Descriptive knowledge may be obtained from sensors. They are characterized by 'Reported' knowledge content, so as to acknowledge the potential disparity between the virtual and real domains (table 2).

Descriptive content for built, assets may be found in proprietary or open schema BIM models and in specifications, drawings and schedules. The current generation of BIM authoring tools make it relatively difficult to rationalize or generalize content, preferring to create large numbers of instances of spaces, components and tasks, even if there is considerable repetition and duplication in shape, naming and attributes. RASE can be used to document an outcome of automatically segmenting and classifying model content, for example identifying the commonalities at the level of system, zone, space type and product type. This content is then closer in presentation to specification documentation which can be reviewed by domain experts for unintended variations. Table 12 expresses the knowledge that "Spaces described as Corridors have Carpet as Flooring except the Reception."

Table 12: Example of descriptive knowledge

RASE type	Metrics
Application	Entity is Space
Application	Description is Corridor
Exception	Name is Reception
Report	Flooring is Carpet

4 RASE BASED SERVICES

Combinations of RASE knowledge sources can support a number of services, ranging in complexity from direct translations through to compliance checking. The examples below assume data driven services, but it is possible to envisage one or more of the services to be provided by a human actor willing to share their regulatory, definitive or descriptive knowledge when required.

4.1 *Comparisons*

Knowledge content can be compared systematically against another (figure 3). In the simplest case, two knowledge documents can be compared to identify where they differ, for example by hypothesizing that the results differ and eliminating those sections and metrics that are identical, leaving those that by differing represent significant change. A revised regulation can be compared systematically against its predecessor, even if there has been a substantial reordering or re-writing, so as to identify where different outcomes will arise. Two definitive documents can be analyzed for developments in the agreed vocabulary. Two or more descriptive knowledge documents can be analyzed for discrepancies and differences.

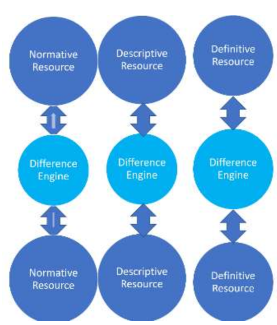

Figure 3: Difference engine to compare resources

4.2 *Guidance and translation*

A descriptive resource can help with explaining other resources, for example the addition of synonyms can help makes documents more accessible (figure 4). Terms can be translated into another language, allowing a technical report to be generated in a language different from that used by the information authors.

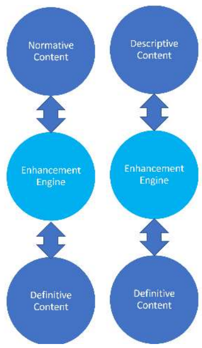

Figure 4: Enhancement engine to expand resources

4.3 *Compliance*

For compliance checking, a normative content is brought up against up against descriptive content, often with the mediation of definitive content to provide mapping services and to provide semantic enrichment (figure 5). Definitive content may not be necessary if the normative content and the descriptive content use the same concepts and vocabulary. Although each knowledge resource can be accessed locally (AEC3 2022), this section explores the opportunity for distributed services so as to enhance privacy and security for each knowledge resource.

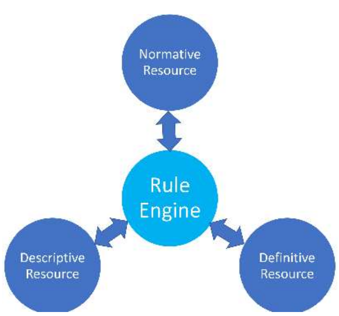

Figure 5: Rule engine for compliance checking

Each knowledge resource may be a document with RASE mark-up or may be other information served up by an application capable of presenting the content as RASE knowledge. We can specify the messages such a service need to respond to, and how the central rule engine can therefore orchestrate an automated compliance checking session, without itself having direct access to each knowledge content. Only information directly relevant to the checking process is requested, and only lowest level entities and values are returned. A RASE knowledge service can be connected to each of the normative, definitive and/or descriptive resources. This also has the advantage of making the core compliance service independent of the three knowledge representations.

Table 12: RASE services

Service	Duties
Rule	Initializes connections to o normative knowledge 'N' o definitive knowledge 'D' o descriptive knowledge 'M' Repeatedly until resolved: o Tells 'N' the previous tfu decision o Asks 'N' for the next relevant metric o Tells 'D' the metric property name o Asks 'D' for the 'M' equivalent o Tells 'M' the previous tfu decision o Asks 'M' for the metric property value o Makes tfu decision by testing metric. o Aggregates tfu decisions upwards
Knowledge 'N', 'D' or 'M'	Accepts connection. Maintains a note of its position Repeatedly to the end of the document: o Notes content of 'tell' from rule engine o Provides the next knowledge packet based on the 'tell' just received.

4.3 Knowledge packet

The concept of a knowledge packet is introduced in the definition of a RASE knowledge service (table 12). This can be thought of and implemented as a single lowest level entity (table 2) found within the knowledge content, such as a metric, term or property. However, when working with distributed services or third party resources it may be optimal to work with sets of such entities, particularly with descriptive resources such as models. This approach requires the transfer of sets of identifiers, rather than a single identifier. An application (Solibri 2022) can perform a geometric check for one instance or for a set of instances, returning lists of those that pass, those that fail and those that are indeterminate (unknown). As the checking process proceeds, some entities will be shown to be not applicable, later some may be shown to not selected, later some may be shown to be excepted before final checks identify those that have passed or failed, or are undecided. Further checking may pick up on an earlier set of entities. The total number of checks performed remains the same but the amount of network traffic and communication overhead is substantially reduced.

5 CONCLUSION

RASE can be used as a mental conceptualization that can help those developing and applying normative content as seen in ISO 12911 (2012). Alternatively, it can be used as a precursor to the generation of other knowledge representations (Nisbet 2022). This paper focusses a third use, the direct exploitation of normative, definitive and descriptive RASE content. Since the RASE mark-up creates a simple hierarchy alongside the document structure, it can be iterated using any depth-first tree iteration algorithm. This can be accelerated by robust heuristics responding to specific events whilst traversing the tree so as to shorten the iterative process. Knowledge resources can be used separately and can be brought together to achieve results such as difference tracking, translation, explanation and most importantly, compliance checking. These experiments have shown that concise algorithms can be used with diverse knowledge resources, but it has not shown that all such resources have a RASE representation. This is the subject of further research.

This suggests that adopting a unified approach to knowledge representation using robust and concise algorithms can generate confidence and trust, particularly for the built environment where confidence in secure, repeatable and reliable performance is persistently low.

6 REFERENCES

AEC3 RASE standard 2021: http://www.aec3.eu/require1/Help_en-GB/help_en-GB_200.html .

AEC3 Require1 2022: http://www.aec3.eu/require1/Help_en-GB/help_en-GB_300.html .

Beach TH, Kasim T, Li H, Nisbet N, Rezgui Y, "Towards automated compliance checking in the construction industry", Lecture Notes in Computer Science , 8055 (2013) 366-380 ISSN 0302-9743 10.1007/978-3-642-40285-2_32

BRE 2018. Building Research Establishment BREEAM New Construction 2018 (UK).

ISO/TS 12911:2012 Framework for building information modelling (BIM) guidance https://www.iso.org/standard/52155.html .

ISO 16739-1:2018 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema https://www.iso.org/standard/70303.html .

Nisbet N, Wix J and Conover D. 2008. "The future of virtual construction and regulation checking", in Brandon, P., Kocaturk, T. (Eds),Virtual Futures for Design, Construction and Procurement, Blackwell, Oxfordshire. doi: 10.1002/9781444302349.ch17

Nisbet N., Ma, L. 2022. Presentations of RASE knowledge mark-up. 2022 European Conference on Computing in Construction. Rhodes, Greece.

NSW 2008. NSW State Environmental Planning Policy (Exempt and Complying Development Codes) 2008. Available: https://legislation.nsw.gov.au/view/html/inforce/current/epi-2008-0572 [Accessed 2022-02-02].

Solibri. 2022. Solibri Model Checker [Online]. Available: https://www.solibri.com/solibri office [Accessed 2022-02-02].

Uniclass 2022 SL table v 1.7 Available: https://toolkit.thenbs.com/uniclass/sl/ [Accessed 2022-02-02].