Implicit Requirements for Ontological Multi-Level Types in the UNICLASS Classification

Chris Partridge
BORO Solutions Ltd
University of Westminster
0000-0003-2631-1627

Andrew Mitchell
BORO Solutions Ltd
0000-0001-9131-722X

Marco da Silva
BORO Solutions Ltd
0000-0001-5007-7995

Oscar Xiberta Soto
BORO Solutions Ltd
0000-0002-0324-0726

Matthew West
Information Junction Ltd
0000-0001-8387-0682

Mesbah Khan
OntoLedgy Ltd
University of Westminster
0000-0003-2631-1627

Sergio de Cesare
University of Westminster
0000-0002-2559-0567

ABSTRACT

In the multi-level type modeling community, claims that most enterprise application systems use ontologically multi-level types are ubiquitous. To be able to empirically verify this claim one needs to be able to expose the (often underlying) ontological structure and show that it does, indeed, make a commitment to multi-level types. We have not been able to find any published data showing this being done. From a top-level ontology requirements perspective, checking this multi-level type claim is worthwhile. If the datasets for which the top-level ontology is required are ontologically committed to multi-level types, then this is a requirement for the top-level ontology. In this paper, we both present some empirical evidence that this ubiquitous claim is correct as well as describing the process we used to expose the underlying ontological commitments and examine them. We describe how we use the bCLEARer process to analyse the UNICLASS classifications making their implicit ontological commitments explicit. We show how this reveals the requirements for two general ontological commitments; higher-order types and first-class relations. This establishes a requirement for a top-level ontology that includes the UNICLASS classification to be able to accommodate these requirements. From a multi-level type perspective, we have established that the bCLEARer entification process can identify underlying ontological commitments to multi-level type that do not exist in the surface linguistic structure. So, we have a process that we can reuse on other datasets and application systems to help empirically verify the claim that ontological multi-level types are ubiquitous.

CCS CONCEPTS

• General and reference • Cross-computing tools and techniques • Empirical studies

KEYWORDS

UNICLASS, top-level ontology, higher order types, first class relations, bCLEARer approach

ACM Reference format:

Chris Partridge, Andrew Mitchell, Marco da Silva, Oscar Xiberta Soto, Matthew West, Sergio de Cesare, Mesbah Khan. 2020. Implicit Requirements for Ontological Multi-Level Types in the UNICLASS Classification. In *Proceedings of the ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems*, 8 pages. https://doi.org/10.1145/3417990.3421414

MODELS '20 Companion, October 18–23, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-8135-2/20/10 $15.00

1 Introduction

In the multi-level type modeling community, claims that most enterprise application systems use ontologically multi-level types are ubiquitous [6]. To be able to empirically verify this claim one needs to be able to expose the (often underlying) ontological structure and show that it does, indeed, make a commitment to multi-level types. Most application systems use linguistically single level types, so prima facie they don't use multi-level types. But the claim is not that their surface linguistic type structure is multi-level, but rather that their underlying ontological type commitment is. To be able to show this, these type commitments must be revealed, and examined to see whether they are, as claimed, multi-level. We have not been able to find any published data showing this being done (though we have abundant evidence in our work on legacy re-engineering these systems – just this has not been published).

From a top-level ontology requirements perspective, checking this multi-level type claim is worthwhile. If the datasets for which the top-level ontology is required are ontologically committed to multi-level types, then this is clearly a requirement for the top-level ontology.

In this paper, we present some empirical evidence that this claim is correct by describing the process we used to expose the underlying ontological commitments and examine them.

The UK's National Digital Twin programme (www.cdbb.cam.ac.uk/what-we-do/national-digital-twin-programme) is in the process of developing a an Information Management Framework (IMF) for a National Digital Twin (NDT) for the UK [2]. This is run by the Digital Framework Task Group (DFTG), which is supported (financially and otherwise) by the UK's Department for Business, Energy & Industrial Strategy (BEIS), Construction Innovation Hub (CIH – www.constructioninnovationhub.org.uk/) and Centre for Digital Britain (CDBB - www.cdbb.cam.ac.uk). Within the IMF there is a FDM Seed project [5]. One component of this is developing a top-level ontology (TLO) to support the NDT domain – and a first stage of this is understanding what the requirements for the TLO would be

The NDT will contain data about, among other things, the critical infrastructure of the UK. The UNICLASS Classification is a comprehensive unified classification system used across the UK construction industry. It is a requirement in the UK, through the standard BS EN ISO 19650 Part 2 National Annex, for BIM (Building Information Modeling) projects to use this. So, the NDT is likely to contain, if not the UNICLASS classification itself, then something very similar. Hence, any underlying high-level ontological commitments made by UNICLASS will also be requirements for the NDTs top-level ontology.

This paper is based upon a project whose goal was to understand the underlying ontology of the UNICLASS Classification. There was a focus on identifying general ontological requirements. This was done with a view to identifying requirements for the NDT top-level ontology.

The project used the first few stages of the bCLEARer™ approach ([3] and described below) to reveal the underlying ontology. This has exposed a couple of general ontological requirements associated with classification – including:

- Higher level (multi-level) types [1] – such as classification and rank
- First-class relations [6] – such as the inter-rank relations

This paper describes the analysis process and these two general ontological requirements.

The body of the paper has the following five sections. Section 2 gives the background to the Project. The Section 3 describes the preparation for the bCLEARer mining process – its COLLECT and LOAD stages. Section 4 describes the analysis – in bCLEARer terms, the EVOLVE – entification process. Section 5 describes the results of the analysis. A final summary section concludes the paper.

2 Background

This section gives the background for:

- The IMF's top-level ontology
- The UNICLASS classification
- The bCLEARer™ approach

2.1 The IMF's Top-Level Ontology

As noted above, the DFTG's NDT programme is in the process of developing an Information Management Framework. They have already developed the Gemini Principles [2] which set out the guiding values for the creation of a (national) system for connecting digital assets. Based upon this, they have developed an approach documented in *A pathway towards an Information Management Framework – A "Commons" for Digital Built Britain* [5]. In this, they explain that an appropriately functioning framework, one which allows digital twins to connect, should include a Foundation Data Model (FDM) which would include the top-level categories and data structures to support the data requirements for the widest range of Digital Twins. They are developing a top-level ontology (TLO) as the top layer of this.

2.2 The UNICLASS Classification

UNICLASS is a dynamic and unified classification system for the construction industry covering all sectors. It is a consistent classification structure for all disciplines in the construction industry. It is a way of identifying and managing the vast amount of information that's involved in a project, and using it is a requirement for BIM (Building Information Modelling) projects, to comply with BS EN ISO 19650 series of standards.

UNICLASS 2015 (https://www.thenbs.com/our-tools/uniclass-2015), the latest version, is divided into a set of tables which can be used to categorise information for costing, briefing, CAD layering, annotations, etc. as well as preparing specifications or other production documents. It contains tables classifying items within a range of scales; from a large facility such as a railway, down to products such as a CCTV camera in a railway station. The classifications within the tables allow buildings, landscape and infrastructure to be classified under one unified scheme.

2.3 The bCLEARer™ Approach

The bCLEARer approach is the latest incarnation of a approach for mining ontologies from legacy systems – that was initially developed in the late 1980s and described in [7]. It has been in continuous development since then [3]. It is one of very few legacy re-engineering approaches, and the only one that focuses on ontology [4] – a key factor here as we wish to identify ontological commitments.

bCLEARer has five levels that can be understood in terms of the standard ascending levels of semantic maturity (see Figure 1).

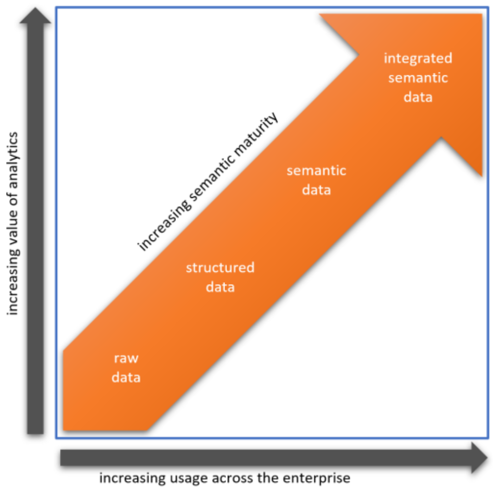

Figure 1: Levels of semantic maturity

The first four stages map onto the standard semantic maturity levels - this mapping is shown graphically in Figure 3. The five stages to the approach are listed in Table 1. Figure 2 gives a picture of its typical processes.

Stages	**Aim**	Maturity Level
COLLECT	Select the data; establish the broad scope	raw data
LOAD	Structure the data	structured data
EVOLVE	(Foundationally) ontologise the data	semantic (foundationally ontologised) data
ASSIMILAT E	Integrate into the global repository	integrated data
REUSE	Publish data in reuse format	

Table 1: bCLEARer stages – and their semantic maturity

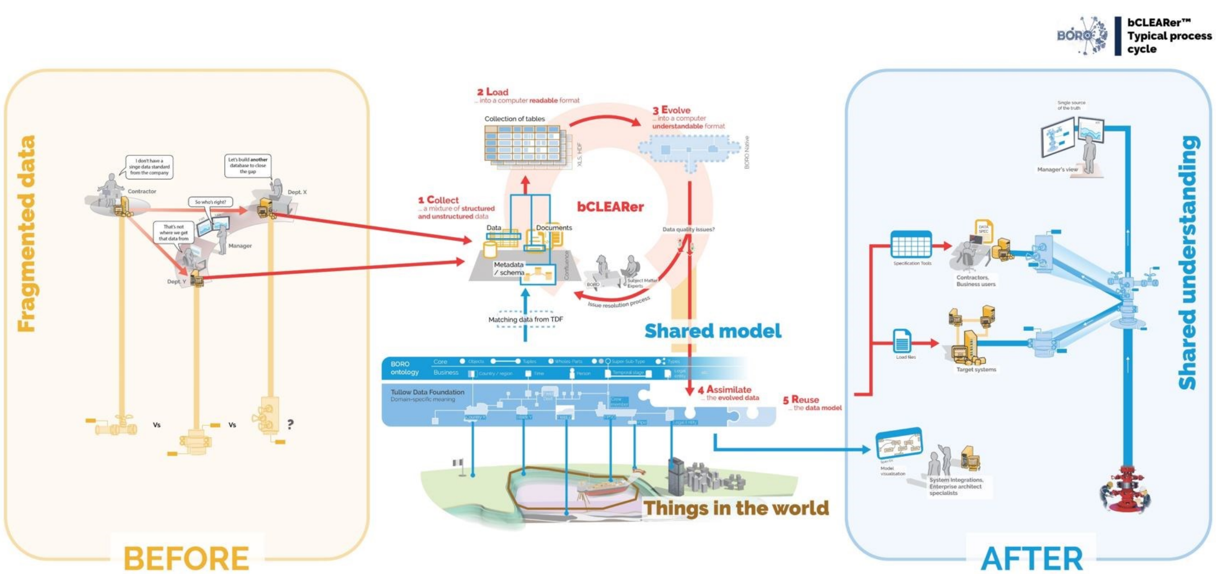

Figure 2: bCLEARer™ Approach

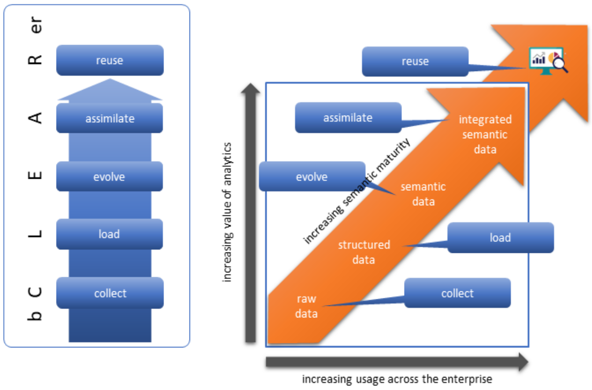

Figure 3: Mapping onto levels of semantic maturity

The EVOLVE stage has three broad sub-stages; entification, object orientation and ontologisation – see Figure 4. For our purposes here, we only undertook the first entification stage, as this exposes many of the general ontological requirements.

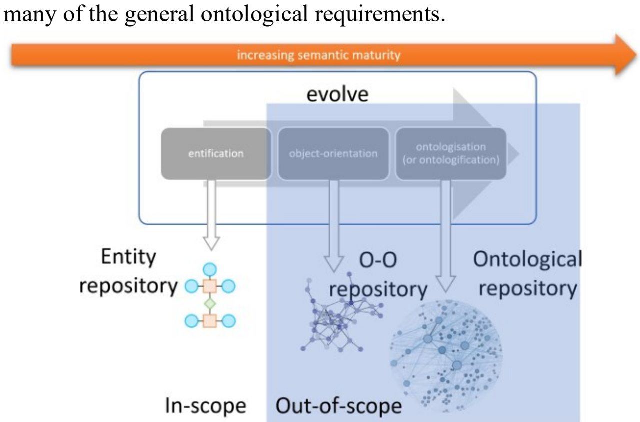

Figure 4: EVOLVE sub-stages

3 Preparing to Mine UNICLASS's Semantics

The first two stages of bCLEARer (COLLECT and LOAD) focus on getting the data ready for processing. A key goal of these stages is getting the data into the row and column format of the table paradigm – as shown in Figure 5.

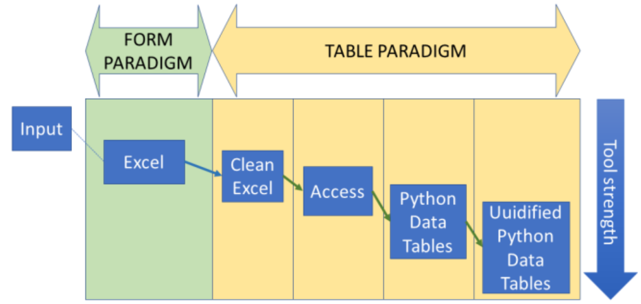

Figure 5: From form to table paradigm

3.1 COLLECT Stage

We collected and stored the twelve spreadsheets from the UNICLASS website. Each of the spreadsheets contains the classifications for an area.

3.2 LOAD Stage

Excel users seem to have an infinite supply of ways to re-organise the data so that it is no longer exactly in a table row and column structure – making it easier for humans to read. We refer to the format of this re-organised data as in the form paradigm. In the LOAD stage, we wind the structure back to a straight-forward row and column structure – that is easier for computers to recognize what is in which column and cell. We refer to the format of this unwound data as in the table paradigm. In this case, each original spreadsheet has effectively one row of a table and a second table underneath it – the first is the first title row of each sheet and the other starts on the third row. This requires some data wrangling. There are three main stages. In the first, we take the first row out from each spreadsheet and reformat and store it in a new title sheet – in a proper table row and column format – see Figure 6.

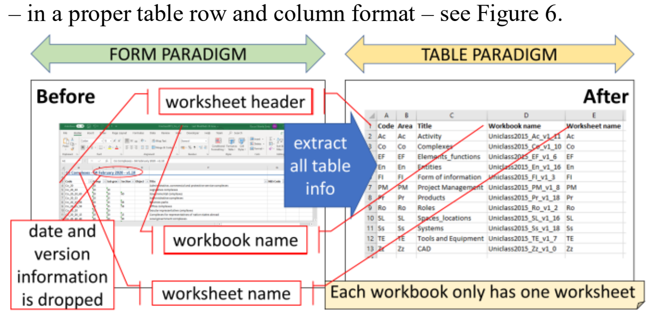

Figure 6: Extract first rows to a new table

In the second, we remove the first two title rows of the original spreadsheets – see Figure 7.

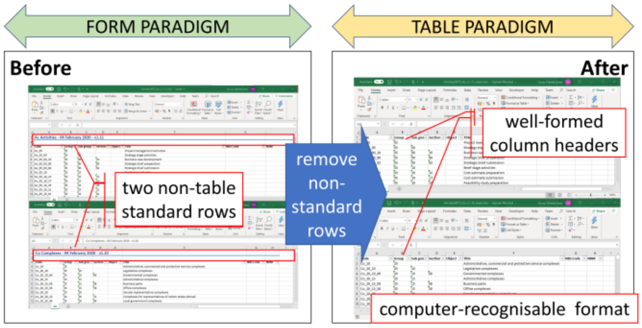

Figure 7: Remove first two title rows

4 Entification – Mining UNICLASS's Semantics

In the project, we only undertake the first – entification – sub-stage of the EVOLVE stage process. The initial stages are mainly data wrangling; refactoring the existing data and adding inferred data.

4.1 Add Tables as Rows – and Area Column

At the start of EVOLVE, there are two datasets. One table taken from the title rows of the original spreadsheets and then a series of tables taken from the body of the spreadsheets; the titles and classifications datasets.

Our first task is consolidating the content of the titles dataset into the classifications dataset, so we have only one dataset. The title rows are a kind of classification much like the rows of the classifications spreadsheets. (We could also argue that the titles type the classifications, but we will not pursue that avenue here.) So we can treat them as a classification and add each row in the first title dataset to its corresponding table in the second classifications dataset – see Figure 8. Once this is done, the first dataset can be deprecated.

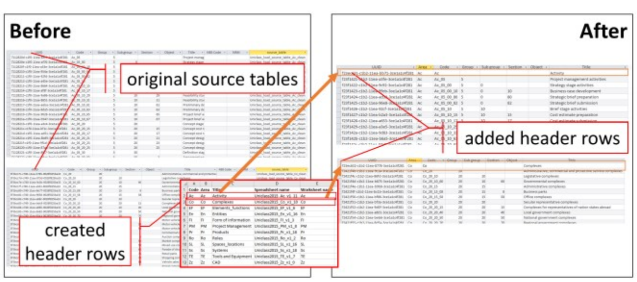

Figure 8: Add header rows to original tables

4.2 Merge Tables

The tables in the classification dataset divide the rows into areas – and so are a kind of proxy for the title rows. Once the title rows are included in the classification spreadsheets, the tables become redundant and so their rows can be merged into a single table – see Figure 9.

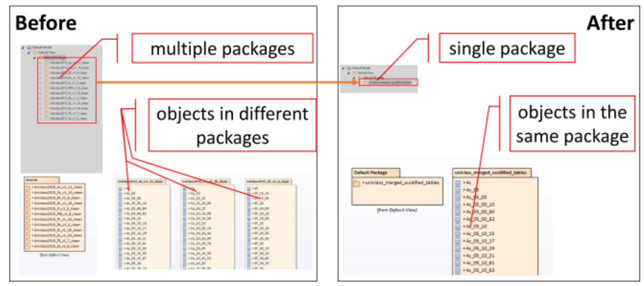

Figure 9: Merging table – visualised in UML

4.3 Extracting the Parent-Child Relations

UNICLASS is a hierarchy, but there are no explicit foreign key links in the data as it stands. The hierarchy can be easily inferred – see Figure 10.

To make the hierarchy explicit we create a parent-child relation table from the implicit data in the table.

Before we add the area title rows, there are 155 separate hierarchies. After we add the area rows and their relations, there are twelve separate hierarchies. We now consolidate these to one, by adding a top element – see Figure 11.

Figure 11 only shows a sample of elements. It is easier to appreciate the consequences from an overall visualisation – **Error! Reference source not found.** shows the consequences of adding areas and the top object.

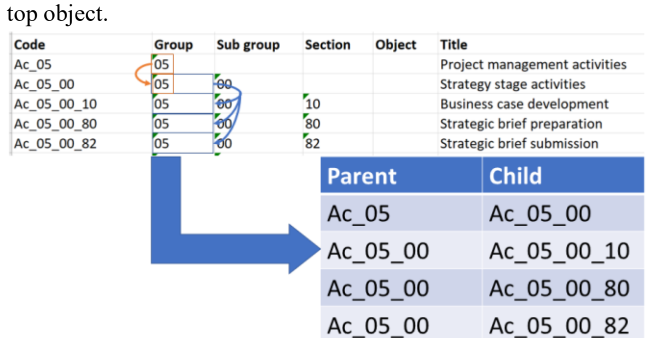

Figure 10: Inferring the parent-child relations

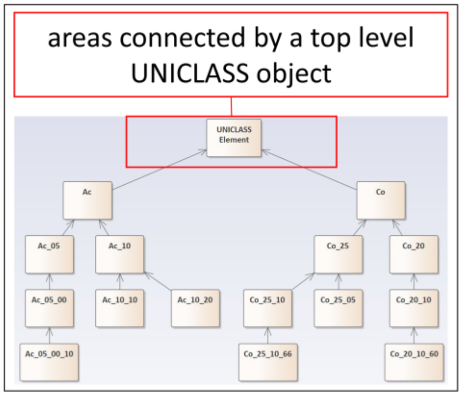

Figure 11: Consolidating into a single hierarchy

4.4 Extracting Ranks

Classifications have a well-studied structure – one of which is the ordering of the classes into ranks [6]. The column headings on the original table (group, etc.) are the ranks of the UNICLASS classification. However, they are only a partial ranking. There are two implicit higher ranks; area and top element. Figure 12 shows how the full set of ranks are extracted. Figure 13 shows a visualisation of the ranks in a UML model.

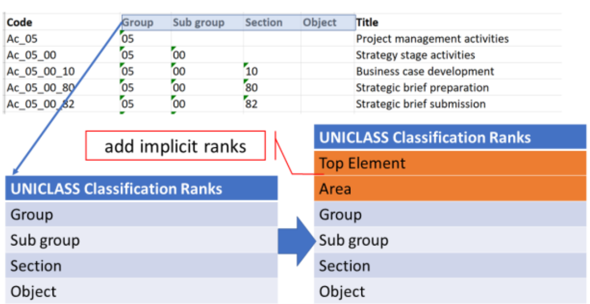

Figure 12: Extracting ranks

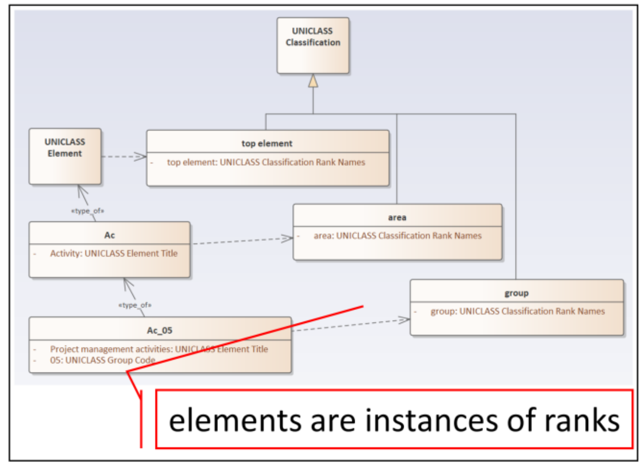

Figure 13: Adding ranks – visualised in UML

Each of the ranks is a UNICLASS rank and is an instance of the type UNICLASS Ranks. We add this type and its instance relations to the data model – see Figure 15.

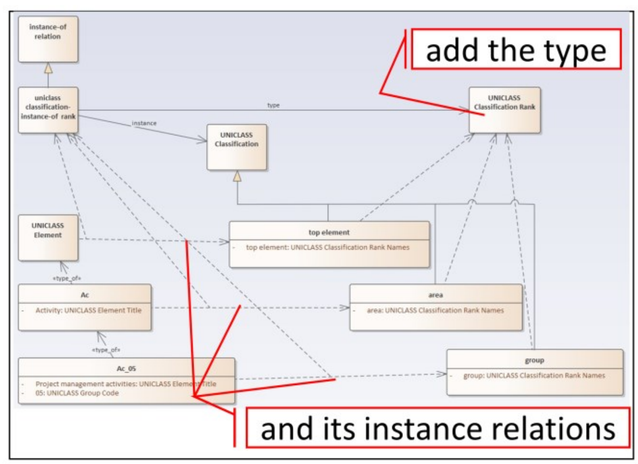

Figure 14: Typing the ranks – visualised in UML

One of the characteristics of ranks in classifications, is that they are disjoint and stratified. A rank will only have children in the next lower rank, no deeper – as shown in Figure 16. These constraints are implemented in the spreadsheets through the way the rank columns are used.

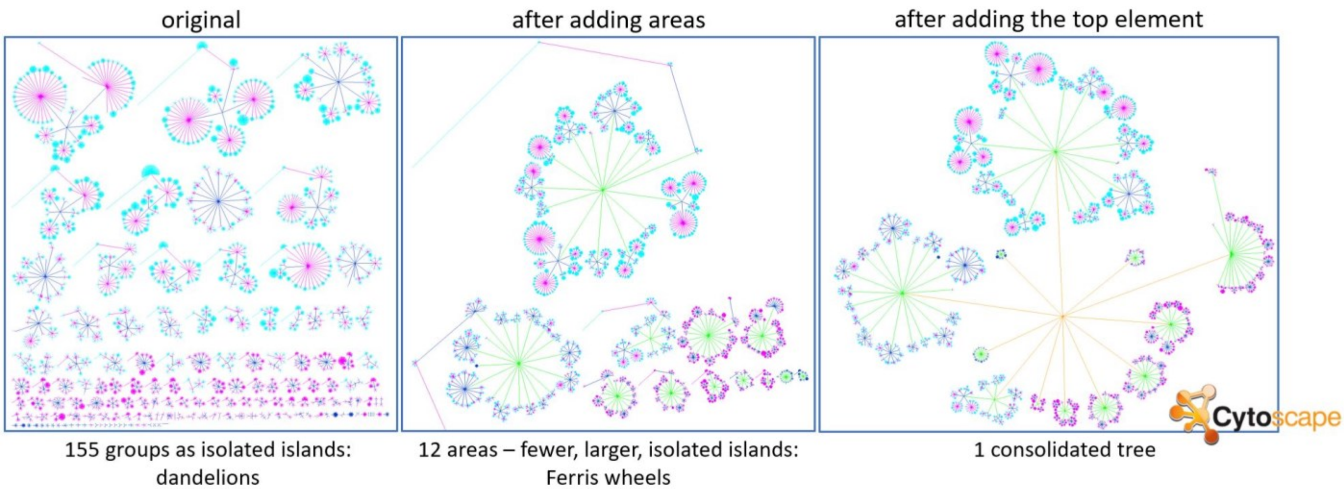

Figure 15: Visualising the consolidation

Code	Group	Sub group	Section	Object	Title
Ac_05	05				Project management activities
Ac_05_00	05	00			Strategy stage activities
Ac_05_00_10	05	00	10		Business case development
Ac_05_00_80	05	00	80		Strategic brief preparation
Ac_05_00_82	05	00	82		Strategic brief submission

UNICLASS Classification Ranks

Rank	Child Rank
Top Element	Area
Area	Group
Group	Sub Group
Sub group	Section
Section	Object
Object	

The table structure ensures the ranks are disjoint.
Each row can belong to one and only one rank

Figure 16 Disjoint ranks

The constraints can be made explicit in the data model by introducing the disjoint relationships between the ranks – as shown in Figure 17.

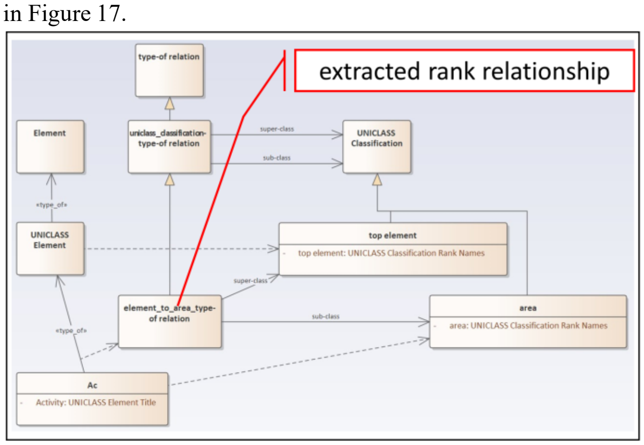

Figure 17: Extracted rank relationships – visualised in UML

4.5 Add Top Level

Finally, the various components of the data model are unified through the introduction of a minimal set of top-level objects – as shown in Figure 18.

5 The Emerging General Requirements

There are two emerging general ontological requirements.

5.1 Higher Order Types

The addition of the minimal top level shows clearly that three levels of types (element, element class and element class class) are needed to describe classifications, their individual ranks and the collection of the classifications ranks. This clearly establishes the existence of ontological multi-level (higher order) types in UNICLASS

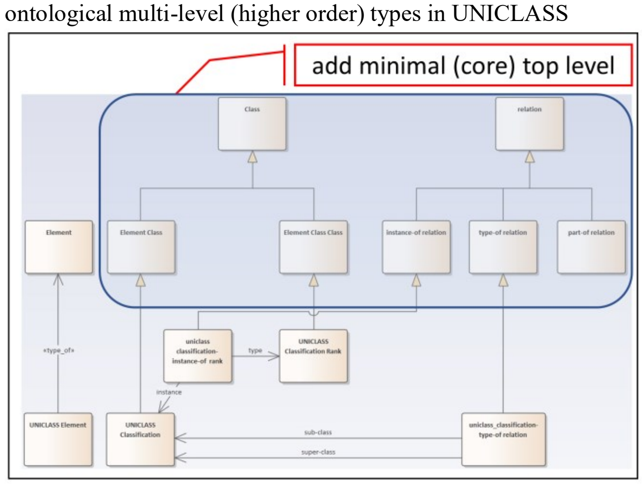

Figure 18: Add minimal (core) top-level

5.2 First Class 'type-of' and 'instance-of' Relations

We introduced relationships between the ranks (see Figure 17) to make the disjoint stratification constraints explicit. These relationships are sub-types of the type-of relation (also known as the super-sub-type relation). We also introduced the 'uniclass classification instance-of' relation to capture the ranks-have-classifications-as-instances pattern (also Figure 17). This is a sub-type of the instance-of relation. In many modelling languages, these relations are not first-class (in the sense of [8]), in that they cannot be sub-types in this way. If we want to capture this constraint and pattern, then the natural way to do requires that these two types of relation are first-class. This establishes the existence of ontological first-class relations in UNICLASS.

5.3 General Ontological Requirements

For any top-level ontology that will include UNICLASS classifications in its data, the identification of these two general ontological patterns implies that it has a requirement to support these.

6 Conclusion

We have described how the bCLEARer entification stages were used to analyse the UNICLASS classifications making their implicit ontological commitments explicit. In particular, we showed how this revealed the requirements for two general ontological commitments; higher-order types and first-class relations. This establishes requirements for a top-level ontology that includes the UNICLASS classification to be able to accommodate these commitments.

From the perspective of the original ubiquity claim, we have established that the bCLEARer entification process can identify underlying ontological commitments to multi-level types that do not exist in the surface linguistic structure. So, we have a process that we can reuse on other datasets and application systems to help empirically verify the claim that ontological multi-level types are ubiquitous.

ACKNOWLEDGMENTS

This work was carried out in collaboration with the two projects.

Firstly, this work was supported by the UK National Digital Twin programme of the Digital Framework Task Group, which is, in turn, supported by the Department for Business, Energy & Industrial Strategy, the Construction Innovation Hub, and the Centre for Digital Built Britain.

And secondly, this work was supported in the project 'Digital Twins in Construction: Towards an Ontological Model Development and Integration Framework' carried out by the Centre for Digital Business Research (University of Westminster, UK) and funded via the Transforming Construction Network Plus. The Transforming Construction Network Plus is funded by UK Research and Innovation (UKRI), an investment supported by the Industrial Strategy Challenge Fund (ISCF)

REFERENCES

[1] Colin Atkinson and Thomas Kühne. 2001. The essence of multilevel metamodeling. In *International Conference on the Unified Modeling Language*, 19–33. DOI: https://doi.org/10.1007/3-540-45441-1_3

[2] A Bolton, M Enzer, J Schooling, and others. 2018. The Gemini Principles: Guiding values for the national digital twin and information management framework. *Centre for Digital Built Britain and Digital Framework Task Group* (2018). DOI: https://doi.org/10.17863/CAM.32260

[3] Sergio de Cesare and Chris Partridge. 2016. BORO as a Foundation to Enterprise Ontology. *Journal of Information Systems* 30, 2 (2016), 83–112.

[4] A. Daga, S. de Cesare, M. Lycett, and C. Partridge. 2005. An ontological approach for recovering legacy business content. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, IEEE, 224a–224a.

[5] James Hetherington and Matthew West. 2020. The pathway towards an Information Management Framework-A "Commons" for Digital Built Britain. (2020). DOI: https://doi.org/10.17863/CAM.52659

[6] Chris Partridge, Sergio de Cesare, Andrew Mitchell, and James Odell. 2016. Formalization of the classification pattern: survey of classification modeling in information systems engineering. *Software & Systems Modeling* (2016), 1–37.

[7] Chris Partridge. 1996. *Business objects: re-engineering for re-use*. Butterworth-Heinemann.

[8] Christopher Strachey. 2000. Fundamental concepts in programming languages. *Higher-order and symbolic computation* 13, 1-2 (2000), 11–49. DOI: https://doi.org/10.1023/A:1010000313106