Towards a human-centred framework for representing and reasoning about social values in buildings

Carl Schultz, Yazan N. H. Zayed, Anna Elisabeth Kristoffersen, Gustaf Lohm, Aliakbar Kamari

Aarhus University, Denmark

Abstract. Promoting the well-being of building occupants is a critical part of ensuring the sustainability of the built environment (in the full sense of Life Cycle Sustainability Analysis, LSCA), pertaining to aspects such as air quality, view outlook, the sense of privacy, thermal comfort, and so on. We present the first prototype version of a new human-centred framework that captures key aspects of the underlying concept structure of qualitative social criteria that are found in popular building certification systems such as the Danish DGNB, WELL Building Standard, LEED, and BREEAM international. We are developing the framework using real human-centred principles taken from literature in psychology, architecture research, DGNB and interviews with architects. We demonstrate a proof-of-concept implementation of the framework on rules about spaciousness via embodied visibility.

1 Introduction

Building certification systems such as DGNB, LEED, and BREEAM can be used to assess the sustainability of buildings, typically based on environmental, economic and social values. The assessment of social values includes aspects such as the sense of privacy, speech intelligibility in different room types, the promotion of interaction between building users, and general acceptance and liking of the building. We seek to develop software tools that can support decision making during the design process by automatically assessing social values, e.g., for assessing, measuring, simulating, analysing, or computing design alternatives and decision impacts. However, developing such software tools is challenging due to the inherent performance-based, qualitative, and subjective character of design criteria pertaining to social values [1, 2, 3].

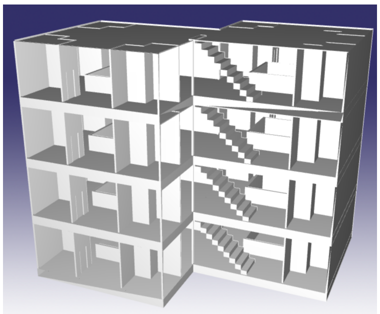
Consider the following arguments that exhibit a kind of *logic of architectural qualities* for privacy and spaciousness: "A tree positioned in front of the window increases the sense of privacy", "the observer has an enhanced sense of spaciousness when the volume of visible space below them is much greater than the volume of visible space above". We would like to develop decision-support software tools that can be programmed with such qualitative human-centred architecture rules and principles. Subsequently, given a BIM model, the tool should use these rules to reason about the predicted subjective impressions of occupants, and thus ultimately support design teams in achieving building sustainability goals in a holistic way.

Key research questions that we address in this paper are:

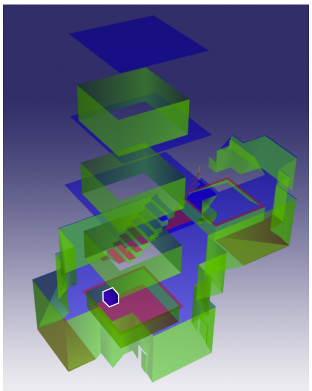
- *Research Question RQ1*: how can diverse principles at different levels of abstraction be unified within one knowledge base? That is, human-centred qualities refer to diverse aspects such as feelings of belonging, well being, happiness, privacy, spaciousness and so on, and these subjective impressions are affected by the spatial layout of building elements, materials, lighting, etc.

- *Research Question RQ2*: how can one reason in a sound and logical way about subjective impressions when the corresponding principles are vague, qualitative and potentially contradictory?

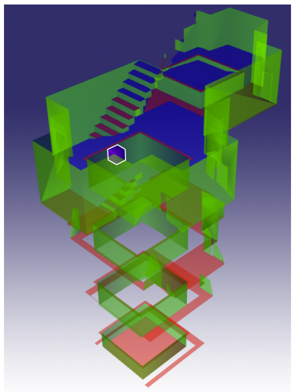
To address these research questions we have developed a human-centred software framework, with two key contributions and novelties:



(a) Office building.



(b) Visibility from lower position: not spacious.



(c) Visibility from Higher Position: Spacious.

Figure 1: Example illustrating embodied 3D visibility analysis.

- *Contribution C1*: we adapt the four *requirement levels* of Lauesen [4] in requirements engineering to define three human-centred formula levels (addressing RQ1)
- *Contribution C2*: rather than treating subjective impressions as logically deduced facts, the framework produces *arguments* for and against those impressions (addressing RQ2).

Together, contributions *C1* and *C2* enable an architect's *design intent* to be formally represented in a way that can be reviewed, verified, and tracked as the BIM model changes in the hands of other specialists in the design team.

1.1 Spaciousness via Embodied 3D Visibility

We will use the following case as a worked example throughout the paper to explain the aspects of our framework and how our framework is used [5]. Embodied 3D visibility is an approach to predicting the visual experience of space that takes the orientation of the observer into account, specifically by distinguishing between the visible space above the observer (ceilings), below the observer (floors), and horizontal visible space (walls).

Through a study with 30 participants, the authors determined that the sense of *spaciousness* corresponds to situations where the volume of visibility below the observer was greater than the volume of visibility above the observer, and where the vertical visibility space (ceilings and floors) had low "jaggedness", i.e. relatively low volume of visible space compared to the visible surface area.

For example, consider the office building illustrated in Figure 1a. Figure 1b illustrates embodied visibility analysis where the observer is near the bottom of the office building (green surfaces are visible walls, red surfaces are visible floors, and blue surfaces are visible ceilings, and the small purple box depicts the location of the observer). Being positioned near the bottom, the observer has more visible space above them rather than below, and participants reported a decreased sense of spaciousness compared to Figure 1c where the observer is positioned near the top of the office building.

1.2 Methodology

We are developing our human-centred framework in an incremental, iterative way, with rounds of interaction with professional architects. The framework is currently in an early stage of development. We have developed a proof-of-concept implementation in Prolog, to help drive the development of the framework and to demonstrate its core features. To develop the ontology to represent architecture quality principles we follow the IDEF5 process for ontology capture (organising and scoping, data collection, data analysis, initial ontology development, refinement and validation). In addition we are engaging in real-world deep renovation projects that are taking place at Aarhus University as part of a large construction programme Campus 2.0, with a focus on reasoning about the perception of air quality.

With respect to data collection and analysis, we have collected principles that cover a broad range of sensory modalities (sight, hearing, touch, smell, movement, affordance, etc.). Sources include well-established building certification systems, and academic sources taken from architecture research and psychology. For each document, we review and record the soft human-centred values that are in focus (privacy, spaciousness, etc.), along with any informal and formal definitions presented, e.g., paraphrasing Flynn in lighting research [6]: *"clarity" is the ability of occupants to discern details in central work areas and along the perimeter of the room.* Table 1 lists key references that we have used intensively thus far.

Table 1: Selection of key references studied for developing the human-centred framework.

Reference	**Target influences**	**Sense modalities**
[5]	(goal) spaciousness, complexity	Visibility
[6]	(goal) clarity, spaciousness, relaxation, intimacy, pleasantness	Lighting, office environment
[7]	(domain) ambient illumination, colour temperature, contrast, hierarchies of emphasis	Lighting
[8]	(goal) view out quality, perceived shape of space (closure, convexity, concavity), complexity, privacy, perceived density	Residential
[9]	(goal) chaotic, calm	Acoustics

2 Related Work

Broadly, our research falls within the area of automated building code compliance checking [10, 1] and building code formalisation [11]. Our particular focus and novelty is on formalising and checking qualitative, performance-based criteria that are "human-centred" (i.e. architectural qualities, or social criteria). Our approach is characterised as being based on first-order logic rules with BIM integration.

Factors of social criteria addressed in building certification systems are described using terms including indoor environmental quality, comfort, and well-being [12, 13, 14, 15, 16, 17]. In this paper, we focus on visibility and the consequent sense of spaciousness.

Numerous other human-centred impressions are based on visibility such as complexity and privacy from outside the building, that is, the ability to remain unseen or unobserved. [8, 18] define four spatial quality determinants that are used to assess the renovation of residential buildings, these determinants are view, internal spatial arrangements, transitions between public and private spaces, and perceived built human densities. Relevant to the sense of privacy, [19] define the privacy parameter of a point in a specific area observed from external spaces, and a visual openness parameter.

3 The human-centred framework

Our human-centered analysis framework consists of three modules (Figure 2): knowledge representation, substitutions and interpretations, and reasoning services. The *knowledge representation* module consists of an ontology of human-centred principles that declares the set of spatial artefact concepts in use, a set of spatial prepositions, and the formalisation of architectural principles in the form of a human-centred knowledge graph.

The role of the knowledge graph is to formally capture architectural principles as logical *formulas* built from the geometric representation of BIM products (e.g. walls, slabs, etc), spatial artefacts, and spatial prepositions.

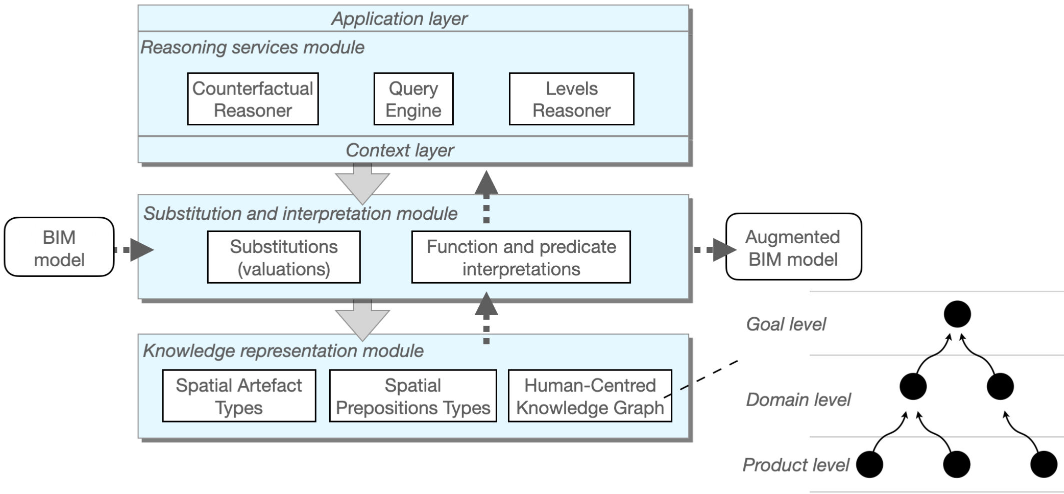


Figure 2: Our Human-Centred Framework, Organised into Three Modules. Large grey arrows indicate flow of control, and dotted arrows indicate data flow.

3.1 Knowledge Graph formulas

Human-centred Knowledge Graph formulas in our framework are first-order formulas, with the usual interpretation of connectives (and, or, not, etc.). They include BIM model *Products* (including so-called *spatial artefacts*, which are a subclass of Products), with the interpretation that the 3D geometric representation of the Product should be used in place of the Product object, so that geometric operators are applied directly to the geometric representations, such as rotation, intersection, etc.

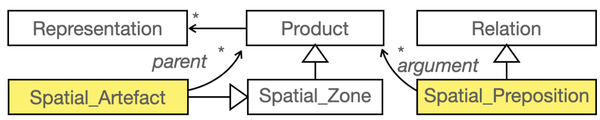


Figure 3: Spatial Artefact and Spatial Preposition classes with standard BIM classes.

Figure 3 illustrates a UML class diagram showing how spatial artefacts and spatial prepositions relate to other classes that are common in standard BIMs such as the Industry Foundation Classes (IFC). The Product class marks the place in the class hierarchy of BIM model objects that can have a geometric representation, e.g. IfcProduct,[1] and typically there is a class such as IfcSpatialZone. Spatial artefacts [20, 21] are a refinement of spatial zones, focusing on occupant experience and behaviour, e.g. visibility, function, movement etc. Spatial artefacts are always *generated* by other products in the building, e.g. signage generates a visibility space where those signs are visible to occupants; we model this relationship as a *parent* attribute between spatial artefacts and their generating products. Spatial prepositions are qualitative spatial relationships between products, e.g. above, between, in front of, etc.

Formally, a first-order *term* is either a variable (with a variable name) or a function (with a function name and a list of zero or more term arguments):

`term` ::= `Var(string)` | `Func(string, list:term)`.

A human-centred (hc) element in a formula is either a Product with a specified class (e.g. "visibility space"), a spatial preposition consisting of a preposition name (e.g. "wide", "high up") and a list of terms, or a predicate consisting of a name and a list of terms:

`hc` ::= `Product(string)` | `Spatial_Preposition(string, list:term)`
| `Pred(string, list:term)`.

[1] This distinguishes from other objects such as "actor", "process", "resource", etc. that do not have a geometric representation. This is what we refer to with the Product class in Figure 3.

A first-order formula ϕ is either true, false, an `hc` element, a connective, or a quantifier:

$$\phi ::= \bot \mid \top \mid \texttt{Atom(hc)} \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi \mid \forall\ \texttt{string} \cdot \phi \mid \exists\ \texttt{string} \cdot \phi.$$

3.2 Knowledge graph levels: goal, domain, product

The formulas in the knowledge graph are organised into three distinct levels based on Lauesen's goal-design requirements scale [4], which we will explain using the spaciousness example via embodied 3D visibility. We formalised these principles of spaciousness via embodied 3D visibility using our own custom *concrete syntax* as follows. Words beginning with uppercase letters are variables. Let P refer to the location of a person, and B refer to a building.

The *goal level* specifies the influence on the occupants that is the ultimate focus of the principle at hand, which in the example case is *spaciousness* at the location of the observer P. This is formalised as a spatial artefact type, e.g. "spacious at" that is derived using the subsequent domain and product level formulas, with the variable P being free in the formula (and ready to be substituted, as opposed to being bound).

```
goal:  spacious_at(P)
```

The *domain level* captures the underlying logic that relates sensory experiences, affordances and behaviour to the goal level item, importantly, without describing *how* the influence is to be achieved by the design of the building. In the example case the domain level formula captures the relationship between visibility spaces using spatial prepositions `much_smaller_than`, `jagged`, the spatial artefact `visibility_space` and functions `above`, `below`, and `vertical`.

```
domain:  much_smaller_than( above(visibility_space(P)),
                            below(visibility_space(P)) ) and
         not(jagged(vertical(visibility_space(P))))
```

The *product level* specifies *how* the domain level effect is achieved in a qualitative way. Returning to the example case, one way of achieving the above domain influence is to position the observer (P) *high up* within the building (B), expressed using the spatial preposition `high_up`:

```
product:  high_up(P,B)
```

Product level formulas do not strictly imply other more abstract level formulas. For example, just because an observer is near the top of a building, this does not imply that the environment necessarily feels spacious. Thus, how can one reason about goal level influences based on product and domain formulas? The key is that the formulas are always applied to a given BIM model, and thus the determination of whether product and domain level formulas hold in the given BIM model is the basis for constructing *arguments* that support goal level architectural qualities. This is handled in the substitution module.

Furthermore, multiple different product formulas can be associated to each domain level formula, and multiple alternative domain formulas can be associated to the same goal level influence. A product level formula can even be directly involved in two contradictory goal levels arguments. For example [8], the installation of balconies in a residential building can decrease the sense of privacy in the adjoining room due to increasing visible exposure from the exterior. On the other hand, balconies also generate a pedestrian *buffer zone* around the balcony, and thus increase the sense of privacy in the adjoining room by creating distance between the room and any people located outside the room that can look in. This example underlines the value in our framework's feature of producing structured *arguments* for, and against, the prediction of architectural qualities, rather than treating architectural qualities as deduced logical facts.

3.3 Substitution and interpretation module

The *Substitution and interpretation module* determines which BIM model objects substitute variables in the knowledge graph formulas, e.g. the substitution module identifies which locations in the BIM model correspond to observer locations (P). It also applies the user-supplied *interpretation* of spatial prepositions, predicates, functions and spatial artefacts, e.g. an interpretation specifies the algorithm for generating the visibility space from a given observer location.

We provide two substitution mechanisms. An *explicit substitution* is an assertion that a given BIM model object unifies with a variable in knowledge graph formula, e.g. in Figure 1b the observer (denoted by the purple box) was asserted to be located at given coordinates. An *implicit substitution* is a "rule" formula such that if a BIM model object satisfies the rule then it can substitute the target knowledge graph variable. E.g. a user may provide a formulas that defines what regions of space qualify as a vantage point for spaciousness analysis.

Formally, a substitution pair is an expression of the form $V = P$ where V is a formula variable and P is a BIM model product. A *substitution* is a set of substitution pairs, $\theta = \{V_1 = P_1, \ldots, V_n = P_n\}$. A substitution θ is applied to a formula ϕ, denoted $\phi\theta$, where each free V_i in ϕ is replaced by P_i. An *explicit substitution* is simply a "substitution" as we just defined, i.e. substitution pairs in the set as specified by the user. *Substitution rules* define a set of substitutions: one substitution for each combination of objects that satisfy the given rules.

3.4 Reasoning services module

The *query engine* is used to determine which formulas are satisfied by which substitutions, and when set up with substitution rules, is used to explore the building to find situations where certain combinations of objects satisfy particular formulas. For example, in Figure 1b the observer location satisfies neither the product level formula (the observer is not high up) nor the domain level formula (the visibility volume above is much larger than the visibility volume below). In Figure 1c this is reversed where both the product and domain level formulas are satisfied. In Figure 4 (a) the product level formula is satisfied but the domain level formula is not satisfied, and in Figure 4 (b) the product level formula is not satisfied but the domain level is.

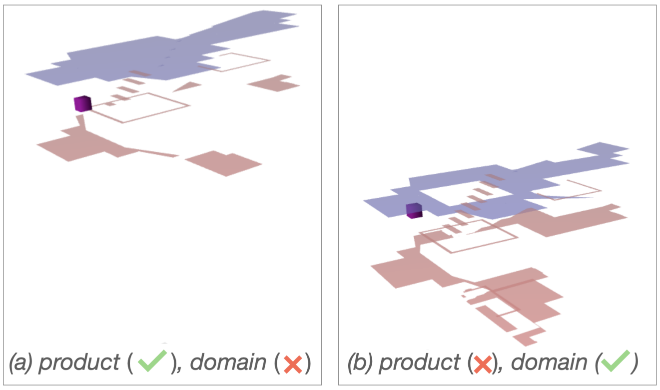


Figure 4: (a) Observer is high up, visibility below is smaller than above; (b) observer is not high up, visibility below is greater than above.

The *levels reasoner* is used to assess arguments for, and against, goal level formulas based on whether product and domain level formulas are satisfied, as summarised in Table 2. In particular, this is critical for verifying *design intent* in the form of a claim: an architect may have specifically designed access to a vista in the office environment to give observers's a sense of spaciousness. However, as the building design in the form of a BIM model is handed to various specialists (structural, fire safety, electrical, HVAC, etc.), numerous changes are often made to the design. Using our framework, the intended human-centred occupant experiences can be tracked, continually verified, and the relevant stakeholders can be informed in case the intended influence becomes invalidated.

Similarly, the levels reasoner is also used to determine unintended *secondary effects*, for example the observer location that the architect designed to be *spacious* may also feel *complex*.

The context layer configures environmental settings e.g. the time of day, the season, orientation of buildings, properties of the observers, etc. with the effect of changing the geometry of the spatial artefacts and the interpretation of spatial prepositions. For example, the user supplies an appropriate interpretation for *visually impaired observers*, *wheelchair user*, etc. for the context layer to deploy.

Table 2: Levels reasoner inferences about goals based on product and domain formulas.

Product formula satisfied	Domain formula satisfied	Goal formula inference	Example
Yes	Yes	Argument exists to support the goal	Figure 1c
No	Yes	Incomplete argument, no explanation for how goal is achieved	Figure 4a
Yes	No	Invalid argument, no conclusion about the goal	Figure 4b
No	No	Argument does not apply, no conclusion about goal	Figure 1b

4 Demonstrative Case: Embodied 3D Visibility

We have implemented the core framework in Prolog as a proof-of-concept, and formalised principles from a range of cases. We will now review one of these cases, namely embodied 3D visibility for predicting the sense of spaciousness.

4.1 Abstract Syntax, Substitutions, Interpretations

The concrete syntax shown in Section 3.2 is first parsed into abstract syntax tree form, maintained within a predicate `formula` that represents a node in the knowledge graph. Each such node consists of a unique identifier, the level (goal, domain, product), and the formula in abstract syntax tree form. The `implies` predicate maintains the implication relations between nodes:

```
formula(f1, goal, pred(spacious_at, [var(p)])).

formula(f2, domain,
 pred(and, [
  pred(much_smaller_than, [
     func(above, [func(visibility_space, [var(p)])]),
     func(below, [func(visibility_space, [var(p)])])
     ]),
  pred(not, [pred(jagged, [
     func(vertical, [func(visibility_space, [var(p)])])])])
])).

formula(f3, product, pred(high_up, [var(p),var(b)] ) ).

implies(f3, f2).
implies(f2, f1).
```

Substitutions are a list of pairs of the form `<variable> = <BIM model product>`:

```
Subs = [
  p = person(v(-500.78, 162.069, 479.12)),  % person location
  b = building(500)] % building of height 500
```

Interpretations are code snippets ("body") to be executed when evaluating predicates and functions. Predicates take the form `<head> = <body>`. Functions deliver a result, and thus their interpretations take the form `<head> = <result variable>` ^ `<body>`. In the following, the implementation of the embodied visibility predicates (e.g. `emvis_volume`, etc.) are omitted for brevity:

```
Interpretations = [
  much_smaller_than(X0, Y0) = (
    emvis_volume(X0, VolX0), emvis_volume(Y0, VolY0),
    2 * VolX0 < VolY0),

  visibility_space(person(L1)) = (M1 ^ emvis_visible_surfaces(L1, M1) ),

  above(V2) = (M2 ^ emvis_filter_surfaces(V2, ceiling, M2) ),

  below(V3) = (M3 ^ emvis_filter_surfaces(V3, floor, M3) ) ]
```

The truth of a formula is then evaluated with respect to a given list of substitutions and interpretations using the framework's `holds` predicate:

```
...,
formula(_, product, Formula),
holds(Subs, Interpretations, Formula).
```

The power of this approach is that substitutions and interpretations are treated as "data objects", which can be analysed, saved with a BIM model, and passed to the *holds* predicate dynamically at runtime so that alternative interpretations of the same knowledge graph formulas can be easily and rapidly evaluated. As stated previously, this context layer in the reasoning services module configures environmental settings which "switches" the interpretation of various functions and predicates, having the effect of changing the shapes of spatial artefacts.

4.2 Built-in Predicates for Evaluating Formulas

The following built-in framework predicates provide the core functionality used by the reasoning service modules. Recall that we define terms to be either variables or functions. Let S be a list of substitutions, and let I be a list of interpretations. Variables are substituted, and functions are interpreted, via the `termval/4` predicate.

```
% Substitute P for variable V when V=P exists
termval(S, _, var(V), P) :- member(V=P, S).

% interpret the function by first evaluating its arguments
termval(S, I, func(Name,Args), Val) :-
 maplist(termval(S,I), Args, Args0),
 Head =.. [Name|Args0],
 findall(V, (member(Head = V ^ Body, I), Body), [Val|_]).
```

The following rules implement the `holds` predicate that evaluates whether formulas hold (i.e. are satisfied) with respect to a given list of substitutions (S) and interpretations (I):

```
% evaluate interpreted predicates
holds(S, I, pred(Name, Args)) :-
  maplist(termval(S,I), Args, Args0),
  Head =.. [Name | Args0],
  forany(member(Head = Body, I), Body).

% evaluate standard logical operators
holds(_,_,true).
holds(S,I,not(F)) :- not(holds(S,I,F)).
holds(S,I,and(F1,F2)) :- holds(S,I,F1), holds(S,I,F2).
holds(S,I,or(F1,F2)) :- holds(S,I,F1) ; holds(S,I,F2).
...
```

The *levels reasoner* is implemented as (built-in) predicates that make inferences according to the logic presented in Table 2. For example, as given in the first row of a Table 2, a goal level inference is *valid* with respect to substitution S and interpretation I if both the implying product and domain level formulas are satisfied:

```
inference(valid,S,I,
          argument([product(PLevel),domain(DLevel)]),
          goal(GLevel)) :-
  formula(GLevel, goal, _),
  implies(DLevel, GLevel), formula(DLevel, domain,  DF), holds(S,I,DF),
  implies(PLevel, DLevel), formula(PLevel, product, PF), holds(S,I,PF).
```

4.3 Verifying and Tracking Design Intent

Suppose the architect intends that an observer located at $(-500, 162, 479)$ experiences a sense of spaciousness. This *design intention* can now be represented using the predicate `design_intent`, with substitution for variables P and B, specifying the *argument* in terms of product and domain level formulas, and the intended goal formula:

```
design_intent([p = person(v(-500, 162, 479)),
               b = building(500)],
              argument([product(f3),domain(f2)]), goal(f1)).
```

The design intent is verified with the *levels reasoner*, with respect to interpretation I, by executing the following Prolog query:

```
?- I = [...],
   design_intent(Subs,Arg, Goal),
   inference(valid,Subs,I,Arg,Goal).
```

This verification can be continually re-run as the BIM model undergoes changes to track whether architectural qualities can still be inferred via the intended valid argument.

Our framework thus enables design intent to be represented in three ways simultaneously, and maintained as part of the BIM model, which we refer to as an *augmented* BIM model. Firstly, the `design_intent` fact represents the intention in the form of an *abstract argument* (product and domain level formulas that infer the goal formula), with substitutions that associate the argument with specific objects in the BIM model. Secondly, the intermediate evaluation results of the design intent formula for each subformula are also recorded. This information is stored as a formula tree which can be queried, for example, to visualise the complete `visibility_space` artefacts and the `above` portion of the visibility space (both represented as 3D meshes). Finally, the interpretations, together with the design intent argument, record the actual procedure for recomputing the argument, enabling the analysis to be directly reproduced in a portable way, i.e. the interpretations are to be delivered with the BIM model. This provides a high degree of reproduceability and transparency in the evaluation of qualitative and vague human-centred principles.

5 Conclusions

We have presented an initial version of our human-centred framework for representing and reasoning about architectural qualities (privacy, complexity, spaciousness etc.) in BIM models, and demonstrated its main features with a demonstrative case study of spaciousness via embodied 3D visibility analysis. Our framework consists of three modules (knowledge representation, binding and interpretation, reasoning services). Our human-centred knowledge graph organises architectural quality principles into three levels: goal, domain, product. We presented the role of a *levels reasoner* in establishing viable arguments for inferring architectural qualities.

We are in the process of fully formalising a wide range of architectural quality principles and their associated design intentions that cover a variety of sensory and behavioural modalities. The aim is to develop an evidence-based tool to support decision-making regarding the inclusion of social intentions in building design, and is used to validate, refine, extend, and improve our human-centred ontology and the particular formalisations we have made.

Acknowledgement We gratefully acknowledge the funding support of the European Union Horizon 2020 research project PROBONO under grant agreement no. 101037075.

References

[1] J. Dimyadi and R. Amor, "Automated building code compliance checking–where is it at," *Proceedings of CIB WBC*, vol. 6, no. 1, 2013.

[2] R. Amor and J. Dimyadi, "The promise of automated compliance checking," *Developments in the built environment*, vol. 5, p. 100039, 2021.

[3] Z. Zhang, L. Ma, and T. Broyd, "Towards fully-automated code compliance checking of building regulations: challenges for rule interpretation and representation," in *In Proc. 2022 European Conference on Computing in Construction (EC^3)*, 2022.

[4] S. Lauesen, *Software requirements: styles and techniques.* Pearson Education, 2002.

[5] J. Krukar, C. Manivannan, M. Bhatt, and C. Schultz, "Embodied 3d isovists: A method to model the visual perception of space," *Environment and Planning B: Urban Analytics and City Science*, vol. 48, no. 8, pp. 2307–2325, 2021.

[6] J. E. Flynn, T. J. Spencer, O. Martyniuk, and C. Hendrick, "Interim study of procedures for investigating the effect of light on impression and behavior," *Journal of the Illuminating Engineering Society*, vol. 3, no. 1, pp. 87–94, 1973.

[7] C. Cuttle, *Lighting by design.* Routledge, 2008.

[8] F. Acre and A. Wyckmans, "Spatial quality determinants for residential building renovation: A methodological approach to the development of spatial quality assessment," *International Journal of Sustainable Building Technology and Urban Development*, vol. 5, no. 3, pp. 183–204, 2014.

[9] A. Gisladottir, "Holistic design method for the urban acoustic environment," Master's thesis, Aarhus University, 2017.

[10] C. Eastman, J.-m. Lee, Y.-s. Jeong, and J.-k. Lee, "Automatic rule-based checking of building designs," *Automation in construction*, vol. 18, no. 8, pp. 1011–1033, 2009.

[11] E. Hjelseth and N. Nisbet, "Capturing normative constraints by use of the semantic mark-up rase methodology," in *Proceedings of CIB W78-W102 Conference*, pp. 1–10, 2011.

[12] J. Dillard, V. Dujon, and M. C. King, *Understanding the social dimension of sustainability.* Routledge, 2008.

[13] R. J. Cole, J. Robinson, Z. Brown, and M. O'shea, "Re-contextualizing the notion of comfort," *Building Research & Information*, vol. 36, no. 4, pp. 323–336, 2008.

[14] M. Boström, "A missing pillar? challenges in theorizing and practicing social sustainability: introduction to the special issue," *Sustainability: Science, practice and policy*, vol. 8, no. 1, pp. 3–14, 2012.

[15] A. Feige, H. Wallbaum, M. Janser, and L. Windlinger, "Impact of sustainable office buildings on occupant's comfort and productivity," *Journal of Corporate Real Estate*, vol. 15, no. 1, pp. 7–34, 2013.

[16] S. Colenberg, R. Appel-Meulenbroek, N. R. Herrera, and D. V. Keyson, "Conceptualizing social well-being in activity-based offices," *Journal of Managerial Psychology*, vol. 36, pp. 327–343, 2021.

[17] A. Kristoffersen, "Social sustainability of buildings a comparison of social qualities in building certification systems, and the development of a catalogue for quantitatively assessing and communicating environmental-, economic-, and social impacts of social intents. master thesis," Master's thesis, Department of Civil and Architectural Engineering, Aarhus University, Denmark, 2023.

[18] F. Acre and A. Wyckmans, "Dwelling renovation and spatial quality: The impact of the dwelling renovation on spatial quality determinants," *International Journal of Sustainable Built Environment*, vol. 4, no. 1, pp. 12–41, 2015.

[19] A. Indraprastha and M. Shinozaki, "Computational models for measuring spatial quality of interior design in virtual environment," *Building and Environment*, vol. 49, pp. 67–85, 2012.

[20] M. Bhatt, F. Dylla, and J. Hois, "Spatio-terminological inference for the design of ambient environments," in *9th Conf. on Spatial Information Theory (COSIT)*, pp. 371–391, Springer, 2009.

[21] M. Bhatt, C. Schultz, and M. Huang, "The shape of empty space: Human-centred cognitive foundations in computing for spatial design," in *2012 IEEE symposium on visual languages and human-centric computing (VL/HCC)*, pp. 33–40, IEEE, 2012.