Modelling Form and Function in Architectural Design

Mehul Bhatt, Joana Hois, and Oliver Kutz

Research Center on Spatial Cognition
SFB/TR 8, Cartesium, University of Bremen
Enrique-Schmidt Strasse, 28359 Bremen, Germany
E-mail: {bhatt, joana, okutz}@informatik.uni-bremen.de

Keywords: Architectural Design, Artefactual Function, Modular Ontologies, Qualitative Spatial Modelling

Abstract. Form, function and the relationship between the two are notions that have served a crucial role in design science. Within architectural design, key aspects of the anticipated function of buildings, or of spatial environments in general, are supposed to be determined by their structural form, i.e., their shape, layout, or connectivity. Where the philosophy of form and function is a well-researched topic, the practical relations and dependencies between form and function are only known implicitly by designers and architects. Specifically, the formal modeling of structural form and resulting artefactual function within design and design assistance systems remains elusive.

In our work, we aim at making these definitions explicit by the ontological modeling of domain entities, their properties and related constraints. We thus have to particularly focus on formal interpretation of the terms "(structural) form" and "(artefactual) function". We put these notions into practice by formalising ontological specifications accordingly by using modularly constructed ontologies for the architectural design domain. A key aspect of our modeling approach is the use of formal qualitative spatial calculi and conceptual requirements as a link between the structural form of a design and the differing functional capabilities that it affords or leads to. We demonstrate the manner in which our ontological modeling reflects notions of architectural form and function, and how it facilitates the conceptual modeling of requirement constraints for architectural design.

Keywords: Architectural Design, Functions in Design, Modular Ontologies

1. Introduction

“Form follows Function” [62] and “Ornament is Crime” [44]—these two doctrines have been the cornerstones of the Modernist tradition in engineering design. Restricting the application of this doctrine to the domain of architectural design, the broad interpretation that it leads to is that the structural form, i.e., shape, layout, connectivity, of a building should be primarily (or more rigidly: solely) determined by its practical function or purpose. Much of the literature in the philosophy of design and architecture, and the ensuing debates thereof, have focused on the semantics of functions with respect to design artefacts and the causal link between form and function, stressing the question of whether or not form should, or indeed does, wholly or in part follows function. Our work is motivated by the practical concerns surrounding the formal interpretation of the terms “(structural) form” and “(artefactual) function”, in particular with respect to their applicability in intelligent architectural design assistance systems. We put these notions into practice by formalising ontological specifications accordingly, in particular, by using modularly constructed ontologies [34] for the architectural design domain [6]. A key aspect of our modelling approach is the use of formal qualitative spatial calculi and conceptual requirements as a link between the structural form of a design and the differing functional capabilities that it affords or leads to. In essence, certain structural forms are inherently (in)capable of producing desired effects with respect to a pre-specified
set of requirements conceptually expressed by an architect or a designer. The main objective of formally modelling aspects pertaining to form and function is to ensure that automated reasoning within design assistance systems becomes possible per se. Our research encompasses the following facets:

**Formal Ontological Modelling in Architectural Design.** The ontologies we develop particularly for the architectural domain are based on different standards and methodologies from ontological engineering as well as architectural design tools. Those ontological specifications that describe the structural aspects of a design, i.e., the floor plan and its relevant information, build on a standardised format for building designs, namely the *Industry Foundation Classes IFC* [19]. Ontological specifications that are related to notions of (qualitative) spatial information are based on different formalisms for qualitative representation and reasoning [12]. Terminological ontologies for specifying high-level design constraints formulated by designers, architects, or engineers, are grounded in foundational ontological engineering methods [47, 66]. Substantially, the ontological modelling of these different aspects, i.e., design, form, function, architectural parts and requirements, are tailored to the architectural design domain and relate form and function in terms of their spatial constraints.

**Qualitative Spatial Modelling.** A crucial aspect that is missing in contemporary design tools is support for explicitly characterising functional requirements of a design. Especially when considering the new generation of building automation systems and smart environments, this is very limiting [6]. For instance, although it is possible to model the spatial layout of an environment at a fine-grained level, it is not possible to model *spatial artefacts* such as the *range space* of a sensory device (e.g., camera, motion sensor), which is not strictly a spatial entity in the sense of having a material existence, but needs to be treated as such nevertheless. For instance, consider the following constraint: ‘The motion-sensor should be placed such that the door connecting room A and room B is always within the sensor’s range space’. The capability to model such a constraint is absent from even the most state-of-the-art design tools. Furthermore, conventional design expertise is often driven by experience and intuition, and is concerned more with spatial and structural aspects of the design rather than its functional characterisation. We here augment exactly these structural aspects by using qualitative spatial constraints [11] to model their functional characteristics.

**Representational Modularity and Cross-Domain Functions.** We specify modular ontologies for constructing and applying domain ontologies for the architectural design of buildings. These ontologies are modularly designed in order to achieve not only their thematically adequate formalisation but also in order to provide a function-based interaction and information exchange across these ontological modules. In detail, the modules are combined by using the theory of $\mathcal{E}$-Connections [41] and refinements [40]. $\mathcal{E}$-Connections’ link relations across ontologies are primarily guided by functional characteristics. This kind of formalisation allows in particular to analyse functional requirements of a given architectural design, i.e., a floor plan. Refinements extend and specify particular requirements for a certain architectural design.

**Organisation of the paper.** The rest of the paper is organised as follows: Section 2 illustrates the basic concepts of structural form and artefactual function. Here, we employ simple, yet real design scenarios and requirements in order to exemplify the relationship between form and function. Section 3 elaborates on the use of spatial and description logics as a means to formalise the ‘design semantics’ and high-level conceptual knowledge and requirements pertaining to architectural design. Section 4 focuses on the role of representational modularity in general, ontological modularity in particular, and also on the application of modularity to modelling multiple perspectives as identifiable within architectural design. Section 5 builds on Section 4 and presents an approach for handling representational modularity using the theory of $\mathcal{E}$-connections. Section 6 then presents in detail an exploratory study of utilizing these modelling constructs to capture the real-world examples introduced in Section 2 using modular ontological specifications of different types of conceptual and spatial information. Finally, Section 7 provides the discussion and outlook for the work described in the paper.
2. Structural Form and Artefactual Function in Architectural Design

A crucial element that is missing in conventional architectural design systems pertains to formal modelling, i.e., representation and reasoning over ‘architectural structures’. Formal modelling of the structural form of an environment, and commonsensical reasoning about the differing functional capabilities that it affords or leads to is necessary to ensure that design time objectives indeed met when the design is deployed in reality. In other words, as all architectural design tasks are concerned with a spatial environment, formal representation and reasoning along conceptual and spatial dimensions is essential to ensure that the designed model satisfies key requirements that enable and facilitate its intended function.

2.1. Structure and Function: A Generic Characterisation

For the purposes of this paper, we interpret structural form and artefactual function in the following manner:

**Structural Form.** The structural form of an environment corresponds to the relative arrangement / configuration of spatial entities, artefacts, and anything else—abstract or real—that may be geometrically modelled or interpreted. For the purposes of this paper, the structural form may be interpreted as a constraint network that determines the relative qualitative spatial relationships between the real and artefactual entities contained within a design. The spatial relationships themselves are grounded to the vocabulary of a formal qualitative spatial calculus [11].

**Artefactual Function.** Artefactual function corresponds to the functionality that a particular structural configuration or arrangement affords, produces or leads to. In general, functions essentially correspond to high-level design requirements that are may be ontologically interpreted as sets of constraints within a task-specific design requirement ontology. From the viewpoint of an ontological terminology, they may also be interpreted as concepts with specific relationships / properties within a requirement specification ontology.

In the remaining part of this section, the afore-stated generic interpretation of structural form and artefactual function is illustrated with concrete examples.

2.2. A Design Task

As a basic use case, consider an architect specialising in the design and development of any general building environment. A typical design challenge would be:
Design the layout of an office environment to satisfy structural and functional requirements that collectively aid and complement (and never hinder) the building’s automation systems (monitoring devices, sensors, etc.), and which, by implication, facilitate the intended smartness of such automation systems.

From the viewpoint of the overall design requirements, aspects of this problem explicitly pertain to the functional aspects (e.g., security, privacy, building-automation, accessibility) of the space being modelled, structural code-checking with respect to building regulations, and also possibly specialized client demands. In so far as the scope of this paper is concerned, functional requirements could be categorized as follows:

I. Client Specifications, Expert Knowledge
Certain areas within a building / floor / room should (not) be trackable by sensing devices such as cameras, motion-sensors. As much as possible, the operation of doors should be non-interfering with the functionality of nearby utilities / artefacts. An example follows:

Example 1 (A Sunny Counter) “Place the main part of the kitchen counter on the south and southeast side of the kitchen, with big windows around it, so that sun can flood in and fill the kitchen with yellow light both morning and afternoon”  
(A Pattern Language (p. 916–918) [1])

II. Statutory Requirements
Regional statutory requirements that stipulate structural constraints and other categorical specifications, e.g., as stipulated by disability access codes. An example follows:

Example 2 (Staircase / Treppen) “Steps of a staircase may not be connected directly to a door that opens in the direction of the steps. There has to be a landing between the staircase steps and the door. The length of this landing has to have at least the size of the door width”.

(Bremen (Germany) Building code [10]) – Staircase / Treppen (§35 (10), p. 24)

III. Specialized Requirements
Specialized requirements correspond to those aspects that arise as a direct result of the specialized nature of the environment being designed, e.g., the design of Museums, Court Rooms / Buildings, Airports, Train- Stations and so forth. An example follows:

Example 3 (US Courts Design Guide) The US courts design guide stipulates an elaborate set of requirements, ranging for precise structural specifications to imprecise, fuzzy and sometimes rather vague guidelines bordering along cultural and aesthetic dimensions. Some rather specific examples follow:

Witness-Box Placement: “Witnesses must be able to see and hear, and be seen and heard by, all court participants as close to full face as possible”.

Barrier-Free Accessibility: “Courtroom areas used by the public must be accessible to people with disabilities. Private work areas, including the judge’s bench and the courtroom deputy, law clerk, bailiff, and court reporter stations, must be adaptable to accessibility. While all judge’s benches and courtroom personnel stations do not need to be immediately accessible, disabled judges and court personnel must be accommodated”.

Judge’s Bench Placement: “The height and location of the judge’s bench expresses the role of the judge and facilitates control of the court. Generally, the judge’s bench should be elevated three or four steps (21-24 inches or 525-600 mm) above the courtroom wall.”.
Visibility: “The entrance or entrance vestibule should be clearly visible and recognizable as such from the exterior of the building. The vestibule should be a minimum of 7 feet in depth and able to handle the flow of traffic at peak times.”

(US Courts Design Guide 2007 [65])

Figure 1 is an example schematisation intended to illustrate the different categories of requirement constraints discussed above. It consists of consistent and inconsistent models of the example requirements / scenarios under considerations. The following aspects, marked as [1–4] in Figures 1(a)–1(b), make the plans of Fig. 1 (in)consistent with respect to the following requirements:

– The sensor / camera is placed at a place where a private area such as the wash-room is within its range (No. 1)
– The operating space of the door of the wash-room interferes with the functional area of the wash-sink, and this arrangement is also not conducive, given disability access requirements (No. 2)
– The operation of the main entrance door interferes with the function of the telephone next to it, and from a structural viewpoint, is also not an ideal placement given its proximity to the staircase (No. 3, 4)

Figure 2 is an example design for a court house, adapted from the US Courts Design Guide, which includes this design as an example of the manner in which the main court proceedings area may be designed. The marked regions in Fig. 2 indicate the requirement from Example 3 pertaining to the placement of the Witness Box. Additionally, an ideal location, in terms of privacy, for the Judge’s private room is also indicated. The spatial structure of the both requirements may be interpreted basically in terms of topological and orientational constraints. However, this is further illustrated in Section 6.2.

3. Spatial Logics and Conceptual Modelling for Architectural Design

A variety of logics are useful for modelling architectural design. These include variants and fragments of classical first-order logic, such as Common Logic and Description Logic, and of course many qualitative
and quantitative spatial logics. The most important aspects that need to be covered by such spatial logics are topology, distance, shape and orientation.

For the purposes of this paper, we focus on modelling aspects and restrict ourselves to define in some detail DLs (Section 3.1), standard qualitative spatial calculi such as RCC8 and logics for reasoning over distances, similarities, and orientations (Section 3.2), as well as combinations of such calculi based on the theory of \( \mathcal{E} \)-connections (defined in Section 5).

### 3.1. Description Logics and OWL

The Web Ontology Language (OWL) has been specifically designed for use on the ‘Semantic Web’. It builds on existing web standards such as XML and RDF whilst being semantically grounded in the formal rigour of expressive Description Logics (DL) [33, 32], which we will describe in some more detail in the following.

Signatures of the description logic ALC consist of a set \( A \) of atomic concepts, a set \( R \) of roles and a set \( I \) of individual constants, while signature morphisms provide respective mappings. Models are single-sorted first-order structures that interpret concepts as unary and roles as binary predicates. Sentences are subsumption relations \( C_1 \sqsubseteq C_2 \) between concepts, where concepts follow the grammar

\[
C ::= A | \top | \bot | C_1 \sqcap C_2 | C_1 \sqcup C_2 | \neg C | \forall R.C | \exists R.C
\]

These kind of sentences are also called TBox sentences. Sentences can also be ABox sentences, which are membership assertions of individuals in concepts (written \( a : C \) for \( a \in I \)) or pairs of individuals in roles (written \( R(a,b) \) for \( a,b \in I, R \in R \)). Sentence translation and reduct is defined similarly as in FOL\(^=\). Satisfaction is the standard satisfaction of description logics.

ALC\(^{\mathrm{ms}}\) is the many-sorted variant of ALC. ALCO is obtained from ALC by adding nominals, i.e. concepts of the form \( \{a\} \), where \( a \in I \). Other logics, like sub-Boolean \( \mathcal{E}L \), ALCO or SHOIN\(^N\), are treated similarly. See [45] for a formalisation as an institution.

The (sub-Boolean) description logic \( \mathcal{E}L \) has the same sentences as ALC but restricts the concept language of ALC as follows:

\[
C ::= B | \top | C_1 \sqcap C_2 | \exists R.C
\]

The logic SROIQ [32], which is the logical core of the Web Ontology Language OWL-DL 2.0\(^1\) extends ALC with the following constructs: (i) complex role boxes (denoted by \( SR \)): these can contain: complex

---

<table>
<thead>
<tr>
<th>Constructor</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intersection</td>
<td>( C_1 \sqcap \ldots \sqcap C_n )</td>
<td>Door \sqcap \text{MainEntrance}</td>
</tr>
<tr>
<td>Union</td>
<td>( C_1 \sqcup \ldots \sqcup C_n )</td>
<td>SwingDoor \sqcup \text{SlidingDoor}</td>
</tr>
<tr>
<td>Complement</td>
<td>( \neg C )</td>
<td>( \neg \text{EmergencyExit} )</td>
</tr>
<tr>
<td>Universal Restriction</td>
<td>( \forall R.C )</td>
<td>( \forall \text{has_material} \cdot \text{Material} )</td>
</tr>
<tr>
<td>Existential Restriction</td>
<td>( \exists R.C )</td>
<td>( \exists \text{has_material} \cdot \text{Wood} \sqcup \text{Aluminium} )</td>
</tr>
<tr>
<td>Max Cardinality</td>
<td>( \leq n R.C )</td>
<td>( \leq 5 \text{has_level} \cdot \text{Floor} )</td>
</tr>
<tr>
<td>Min Cardinality</td>
<td>( \geq n R.C )</td>
<td>( \geq 1 \text{has_element} \cdot \text{Door} )</td>
</tr>
</tbody>
</table>

Table 1: Examples for Description Logic Concept Constructors

---

\(^1\)See also http://www.w3.org/TR/owl2-overview/
### Axiom

<table>
<thead>
<tr>
<th>Axiom</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsumption</td>
<td>$C_1 \subseteq C_2$</td>
<td>Wood $\subseteq$ Material</td>
</tr>
<tr>
<td>Equivalence</td>
<td>$C_1 \equiv C_2$</td>
<td>Door $\equiv$ SwingDoor $\sqcup$ RevolvingDoor $\sqcup$ SlidingDoor</td>
</tr>
<tr>
<td>Disjointness</td>
<td>$C_1 \subseteq \neg C_2$</td>
<td>SlidingDoor $\subseteq \neg$ SwingDoor</td>
</tr>
<tr>
<td>Inverse</td>
<td>$R_1 \equiv R_2^-$</td>
<td>has_conceptual_structure $\equiv$ has_metrical_structure^-</td>
</tr>
<tr>
<td>Functional Role</td>
<td>$\top \sqsubseteq \leq 1R$</td>
<td>$\top \sqsubseteq \leq 1$ has_metrical_structure</td>
</tr>
<tr>
<td>Inverse Functional Property</td>
<td>$\top \sqsubseteq \leq 1R^-$</td>
<td>$\top \sqsubseteq \leq 1$ has_metrical_structure^-</td>
</tr>
</tbody>
</table>

Table 2

Examples for Description Logic Axioms

![RCC8 base relations](image)

Fig. 3. The RCC8 base relations.

role inclusions such as $R \circ S \subseteq S$ as well as simple role hierarchies such as $R \subseteq S$, assertions for symmetric, transitive, reflexive, asymmetric and disjoint roles (called RBox sentences), as well as the construct $\exists R$.Self (collecting the set of ‘$R$-reflexive points’); (ii) nominals (denoted by $O$); (iii) inverse roles (denoted by $I$); qualified and unqualified number restrictions ($Q$). For details on the rather complex grammatical restrictions for $SROIQ$ (e.g. regular role inclusions, simple roles) compare [32], and see the example given below.

Apart from some exceptions\(^2\), description logics can be seen as fragments of first-order logic via the standard translation \([2]\) that translates both the syntax and semantics of various DLs into untyped first-order logic.

Tables 1 and 2 illustrate the various complex class constructors and TBox axioms respectively provided by basic Description Logics. All of these are supported by the present version of OWL 2 DL. The examples in the right hand side of the tables illustrate some of the usages we will make of DL expressivity in the modularly defined ontologies specified in Section 6.1.

### 3.2. Qualitative and Quantitative Spatial Logics

**Topology.** The Region Connection Calculus RCC8 [51] is heavily being used in qualitative spatial representation and reasoning, and we will give examples below on how this kind of reasoning can be combined with ontological reasoning, see Sec. 5. Figure 3.2 displays the 8 basic relations of RCC8, which are mutually exclusive and exhaustive in describing the possible overlap and touching relationships between two (well-behaved\(^3\)) regions in space.

To have a slightly more expressive logic at hand in which we can use the RCC8 relations, and which is more straightforwardly used in an $E$-connection setting, it is convenient to encode the RCC8 relations in a

---

\(^2\)For instance, adding transitive closure of roles or fixpoints to DLs makes them decidable fragments of second-order logic [8].

\(^3\)This is typically taken to mean regular-closed subsets of a topological space, i.e. regions $X$ such that $X = CL(X)$. 
topological logic. The modal logic $S4_u$, i.e., Lewis’s modal system $S4$ with the universal modality added, can for instance be used for this purpose. $S4_u$ is complete with respect to a semantics based on topological spaces as the intended interpretation. Here, the propositional variables are interpreted as subsets of a topological space, the necessity operator $\Box$ is interpreted as the closure operator $C$, and the universal quantifier $\forall$ as universal quantification over all points of the topological space. Sentences are built using propositional variables and these three unary modal operators $\Box$, $\Diamond$, $\forall$.

The basic system is taken from [42] which contains a various axiomatisations of these logics. See [46] for modal logics that explicitly introduce modal operators for the eight RCC8 relations.

**Distance and Similarity.** Being able to specify metrical constraints about absolute and relative distance, or more qualitative constraints about relative ‘closeness’ of objects, are quite obviously very important requirements in a spatial design task. We here introduce one such family of logics, namely the distance and similarity logics of [38, 59, 60]. The basic idea here is to augment a structure $W$ with a distance (or similarity) measure $d : W \times W \mapsto \mathbb{R}^+$, which maps pairs $(a, b)$ of elements of $W$ to a positive real number (including zero), called the distance between the points $a$ and $b$. In the context of working in the euclidean plane, $d$ will typically be assumed to be a metric, i.e. satisfying, for all $x$, $y$, $z \in W$, the following axioms:

$$d(x, y) = 0 \text{ iff } x = y \quad d(x, z) \leq d(x, y) + d(y, z) \quad d(x, y) = d(y, x)$$

Here, distance zero means that the objects $a$ and $b$ are located at the same position. In the following we sketch the syntax and semantics of basic distance logics as well as their interpretation as logics for similarity. We will apply these logics in Section 6.

The basic distance logics are syntactically defined just like standard modal logics such as $S4_u$, i.e. we have a family of propositional variables $\{p_i : i < \omega\}$, Boolean connectives, $\land$ and $\neg$, and a list $\{A_{\leq a}, A_{> a}, \ldots : a \in M\}$ of (unary) modal operators depending on a set $M$, called the parameter set, of non-negative real numbers that we allow as parameters $a$ in formulae. Well-formed formulae in this language are now constructed in a the standard way.

Other Booleans as well as the dual modal operators $E_{\leq a}$ and $E_{> a}$ are defined as abbreviations (e.g., $E_{\leq a} = \neg A_{< a} \land$, $E_{> a} = \neg A_{> a} \land$). Models for this logic are of the form $\mathcal{B} = \langle W, d, \phi_0, \phi_1, \ldots \rangle$, where $(W, d)$ is a distance space and the $\phi_i$ are subsets of $W$. The truth-relation $\langle \mathcal{B}, w \rangle \models \phi$ for this language is completely standard except for the distance operators. They can be used to define ‘complex regions’ as follows:

$$(A_{\leq a} \phi)^\mathcal{B} = \{ w \in W \mid \text{ for all } u \in W \text{ with } d(w, u) \leq a \text{ we have } u \in \phi^\mathcal{B} \}$$

Note that this language already allows to define standard modal operators such as the universal modality, the difference operator, as well as nominals [16, 25]. In an obvious way we can also define more complex operators such as $A_{< b}^a$, etc.

Rather than interpreting the measure $d(a, b) = x$ as a metrical distance, it can also be understood as a similarity measure between $a$ and $b$, where $a$ is more similar to $b$ the smaller the measured distance $x$ is.

---

4. When interpreted over standard Kripke-semantics, models are based on frames with reflexive and transitive relations and $\neg$ is universal quantification over worlds.

5. See [46] for modal logics that explicitly introduce modal operators for the eight RCC8 relations.

6. The basic system is taken from [42] which contains a various axiomatisations of these logics.
This interpretation suggests in particular the following binary operator:

$$(C \equiv D)^B = \{ w \in W \mid d(w, C) < d(w, D) \}$$

where the distance $d(w, C)$ is defined as the infimum of the distances $d(w, u)$, $u \in C^B$ between $w$ and members of $C$. This is an interesting operator, studied for instance in [58], allowing for instance to define Voronoi tessellations using prototypes.

Metrical information, of course, is ubiquitous in architectural floor plans, as discussed below in detail. However, ‘unfinished’ design plans are also a good example for a mix of open world/closed world reasoning. I.e. some distances may be fixed, whereas others are left completely open (for instance by using variables for the distances). In a floorplan, some details might be completely determined/specified (so database reasoning can be applied), whereas other aspects are left open. In general, when a floor plan is seen as a model for a distance logic, it corresponds more to a map-like representation with fixed extensions of regions and distances. Such a scenario can be formally realised by using object and region constants with a fixed interpretation (rather than variables)—see [41] for such a scenario applied to a city map.

**Orientation.** For representing relative orientation in recent years many different calculi have been presented, e.g., the DCC [18] and the Dipole Calculus [55]. Here, we apply the $OPRA_m$ approach [49] because of its expressiveness. The calculi in this family are designed for reasoning about relative orientation relations between oriented points (points in the plane with an additional direction parameter) and are well-suited for dealing with objects that have an intrinsic orientation. An oriented point $\vec{O}$ can be described by its Cartesian coordinates $x_O, y_O \in \mathbb{R}$ and a direction $\phi_O \in [0, 2\pi)$ with respect to an absolute frame of reference. With the parameter $m$ the angular resolution can be influenced, i.e., the number of base relations is determined.

In the case of $OPRA_2$, the orientation calculus we apply in our examples, for each pair of oriented points, 2 lines are used to partition the plane into 4 planar and 4 linear regions (see Fig. 3.2). The orientation of the two points is depicted by the arrows starting at $\vec{A}$ and $\vec{B}$, respectively. The regions are numbered from 0 to 7, where region 0 always coincides with the orientation of the point. An $OPRA_2$ base relation is a pair $(i, j)$, where $i$ is the number of the region, seen from $\vec{A}$, that contains $\vec{B}$ and vice versa. These relations are written as $\vec{A} \angle i_j \vec{B}$. Additional base relations describe situations in which both oriented points are at the same position but may have different orientations ($m\angle i$).

**Size, Shape, Morphology, and Spatial Change.** While the most important aspects of space are topology, orientation, and distance, other aspects of space include size, shape, morphology, and spatial change (over time) [53]. For the purpose of this paper, we use spatial logics primarily for topology, orientation, and distance, as they are sufficient for our modelling examples. They are, in particular, available as theories formulated by spatial logics. Classifications of shapes are, for example, available as an ontological specification [27].
4. Ontological Modularity and Multi-Perspective Representation

Modularity has become a key issue in ontology engineering in recent years. Research into aspects of modularity in ontologies covers a wide spectrum. Applied to ontology engineering, modularity is central not only to reduce the complexity of understanding ontologies, but also in maintaining, querying and reasoning over modules. Here, distinctions between modules can be drawn, for instance, on the basis of structural, semantic, or functional aspects. The construction of modular ontological components can then be exploited by composing or combining such modules, or by indicating ‘links’ between such ontologies.

In particular, the reuse and sharing of information and resources encoded in different ontological (or logical) modules depend on purpose-dependent, logically versatile criteria. In our present context, such purposes include ‘tight’ logical integration of different ontologies (wholly or in part), ‘loose’ association and information exchange across different representation formalisms, and alignment of vocabularies.

In this section we will give a birds eye view of problems of ontological modularity, then introduce several relevant formalisms for spatial design in Section 3, and finally go into greater detail on how these formalisms can be combined on the basis of the theory of ϵ-connections in Section 5.

To start our discussion, we formulate three orthogonal questions that define the research area of modularity for ontologies:

– How can large and complex ontologies be built up from parts, possibly being formulated in different logical languages, and in what ways can those parts be related? (modular combination problem)

Conversely, given a large ontology, how can we decompose it into ‘meaningful’ modules? (modularisation problem)

– How can the structure of a modular ontology be represented, and how can various logical (or structural, topical) properties of the parts (modules) be preserved?

– How can we perform (automated) logical reasoning over such structured ontologies, and how, or when, can we reduce reasoning in the overall ontology to the ontology’s component modules?

The main research question is how to define the notion of module and how to re-use such modules. We briefly summarize some of these aspects and present details relevant to the notion of module, modularity, and reasoning problems. We finally outline the kinds of modules and modular reasoning problems that we encounter in architectural design.

4.1. Aspects and Dimensions of Ontological Modularity

The main dimensions of ontological modularity and respective (automated) reasoning challenges are:

The Language Layer and Semantic Heterogeneity. Whenever we want to combine two ontologies (or formal theories), we run into the problem of syntactic and semantic heterogeneity. Indeed, even if we stay in the same formal logic, we run into the problem of reconciling the joint vocabulary of the ontologies. The most general solution to this problem is to provide a family of logic translations that allows to seamlessly move from one logic to another along the translation, based on a general definition of logic and logic translation as provided by institution theory [24]. Tool support for such translations is, for instance, provided by the HETS system [50].

7The book [61] as well as the workshop series on Modular Ontologies [29, 13, 54, 39] give a good overview of the breadth of this field.

8This is, of course, not an exhaustive analysis.
Structuring, Extension, and Refinement. The mere size of ontologies can make the design process quite hard and error prone (at least for humans). This issue has been only partly cured in OWL by the imports construct, which essentially copies the axioms of one ontology into another. Natural operations are, for instance, union, intersection, ‘hiding’ certain symbols, and extension. The semantics, however, of such operations is in general non-trivial. Methods developed for (algebraic) specification, for instance, can be applied to ontology engineering, as they provide systematic structuring techniques [40]. Apart from such structuring concepts, another natural relationship between ontologies is that of a refinement: \( O_2 \) refines \( O_1 \) if all of \( O_1 \)’s axioms are entailed by \( O_2 \) (possibly along a translation). Essentially, this means that we need to provide a theory interpretation of \( O_1 \) into \( O_2 \) [36]. Another kind of ‘extension’ is provided by the idea of concrete domains. They extend an ontology language by constructs that allow to ‘import’ computations in specific structures, such as the natural numbers or time intervals [28].

Logical Independence. One of the most important logical concepts of modularity is given by the notion of conservativity. An ontology \( O_2 \) is a conservative extension of \( O_1 \) if all assertions made in the language of \( O_1 \) that follow from \( O_2 \) already follow from \( O_1 \). Essentially, this means that \( O_1 \) completely and independently specifies its vocabulary, with respect to \( O_2 \). This concept can, for instance, be used to extract logically independent modules from a large ontology. While this notion of module therefore is important, it is also computationally difficult. Although proving conservativity is undecidable for first-order logic and many expressive description logics (DL) [61], there are general algorithmic solutions for less expressive DLs [36]. The simplest case of a conservative extension is a definitional extension, as it extends the vocabulary of an ontology \( O \) by new terms, whose meaning is entirely determined by the axioms given in \( O \).

Matching and Alignment. Matching [17] and aligning [68] ontologies focus on the identification of (thematically) overlapping parts of two ontologies (matching problem) and on systematically relating terms across ontologies that have been identified as, for instance, synonymous (alignment problem). As opposed to structuring and conservativity, such relationships are often established by using statistical methods and heuristics, employing, for instance, similarity measures and probabilities.

Integration and Connection. Informally, an integration of two ontologies \( (O_1 \) and \( O_2 \)) into a third ontology \( O \) is any operation by which \( O_1, O_2 \) are ‘re-interpreted’ from the (global) point of view of \( O \). This has been utilized in the approach of [56] (called semantic integration), which integrates two ontologies by mapping (or translating) them into a common reference ontology. The main feature here is that semantic consequence is preserved upwards to the reference ontology.

Intuitively, the difference between integrations and connections is that in the former, we combine two ontologies \( O_1 \) and \( O_2 \) using an often large and previously-known reference ontology \( O \), where the models of \( O \) are typically much richer than those of \( O_1 \) and \( O_2 \). In the latter, we connect two ontologies in such a way that the respective theories, signatures, and models are kept disjoint, and a (usually small and flexible) bridge theory formulated (in a bridge language) over a signature that goes across the sort structure of the components is used to link together the two ontologies. Connections, in the form of \( E \)-connections, will be introduced in detail in Section 5 and employed extensively for modelling architectural design in Section 6.

4.2. Modularity Specifications for Architectural Design Specification

Multi-Perspective Semantics & Representational Modularity. Consider the illustration in Fig. 5: an abstraction such as a Room or Sensor may be identified semantically by its placement within an ontological hierarchy and its relationships with other conceptual categories. This is what a designer must deal with during the initial design conceptualisation phase. However, when these notions are transferred to a Computer-Aided Architecture Design (CAAD) tool, the same concepts acquire a new perspective, i.e., now the designer must deal with points, line-segments, polygons and other geometric primitives available within the feature hierarchy of the design tool, which, albeit necessary, are in conflict with the mental image and qualitative conceptualisation of the designer. Given the lack of semantics, at least within
contemporary design tools, there is no way for a knowledge-based system to make inferences about the conceptual design and its geometric interpretation within a CAAD model in a unified manner.

The aspects of ontological modularity that we employ in order to realize the envisioned application to architectural design are manual alignments, conservative (definitional) extensions, $\varepsilon$-connecting thematically different ontologies, and global extension.

**Thematic module.** A thematic module for a domain $D$ is an ontology that covers a particular aspect of or perspective on $D$. The main impact of this notion is that we assume that two thematically different modules for $D$ need to be interpreted by disjoint domains. An example, that we will elaborate on later, is the conceptual space of materials of objects and qualitative representations of topological relationships between such objects: these interpretations clearly should not overlap. We specify such thematic modules as modular ontologies that are part of a certain architectural perspective, such as the conceptual or qualitative spatial, as introduced in Section 6.1.

**Definitional Extensions.** New concepts are, for instance, added to the DOLCE-Lite ontology in the conceptual layer ontology (see Section 6.1) by a definitional signature extension. Moreover, we add new concepts to the spatial relations in the ontology of structural building entities, again, in a definitional manner.

**Linking thematic modules.** Alignments are given by the human expert (the architect), identifying certain relationships between thematically different modules. An overall integration of these thematic modules is then achieved by $\varepsilon$-connecting the aligned vocabulary along newly introduced link relations and appropriate linking axioms. Dedicated reasoning support for general $\varepsilon$-connections is not available at the moment. However, when $\varepsilon$-connecting just ontological modules given in $\text{OWL}^*$, reasoning support can still be realized by a complete encoding of the semantics of $\varepsilon$-connections into $\text{OWL}^*$ DL as follows (compare also [14]): (1) disjointness of thematically different domains is enforced by introducing new ‘local’ top concepts for each ontology (2) domain and range of link relations are accordingly restricted; (3) as $\varepsilon$-connection operators we can use DL existential and universal restrictions for these link relations. Fig. 9 below illustrates how the different architectural perspectives are $\varepsilon$-connected with each other.

**Global extensions of integrated representations.** New constraints are added on top of the integrated representation by $\varepsilon$-connections. Moreover, the process of building integrated representations might be iterated at a later stage of the specification process, integrating further ontologies whilst treating the previously built representation as a new ‘monolithic’ building block (see Section 6.2).

---

9 Note that this technique can also be applied to other logics that can be encoded in $\text{OWL}^*$, such as RCC8, compare [67].
We have presented some of the key aspects of ontological modularity in general and summarised the aspects of multi-perspective modelling that we will employ for the domain of architectural design in the remainder of this paper. Next, we will present in some detail the theoretical foundations for multi-perspective modelling based on $\mathcal{E}$-connections.

5. $\mathcal{E}$-Connecting Conceptual and Spatial Dimensions

Heterogeneous knowledge representation was a major motivation also for the design of ‘modular ontology languages’, such as distributed description logics (DDLs, [9]) and $\mathcal{E}$-connections [43, 41]. We here concentrate on the latter. $\mathcal{E}$-connections were originally conceived as a versatile and computationally well-behaved technique for combining logics, but were subsequently quickly adopted as a framework for the combination of ontologies in the Semantic Web [15].

The general idea behind this combination method is that the interpretation domains of the connected logics are interpreted by disjoint (or sorted) vocabulary and interconnected by means of link relations. The language of the $\mathcal{E}$-connection is then the union of the original languages enriched with operators capable of talking about the link relations.

$\mathcal{E}$-connections have also been adopted as a framework for the integration of ontologies in the Semantic Web [14], and, just as DLs themselves, offer an appealing compromise between expressive power and computational complexity: although powerful enough to express many interesting concepts, the coupling between the combined logics is sufficiently loose for proving general results about the transfer of decidability. But as follows from the complexity results of [41], $\mathcal{E}$-connections in general add substantial expressivity and interaction to the components. Here, the transfer of decidability as well as the expressiveness of the obtained $\mathcal{E}$-connection depend not only on the component logics but, essentially, on the employed connecting link language.

In $\mathcal{E}$-connections [41], specifically, a finite number of formalisms, typically talking about distinct domains or distinct views on the same domain, are connected by relations between entities in the different domains, capturing different aspects or representations of the ‘same object’. For instance, concerning the logics that we employ to model architectural design, namely logics for topology, distance, and orientation that we introduced in Section 3, informally, the following link relations are relevant: (1) a concept $C$ of a conceptual module specified in a DL$_1$ can be related via a link relation to a corresponding concept in another conceptual module specified in another DL$_2$—see Example 4 and Figure 6 below; (2) an ‘abstract’ object $o$ of a description logic DL$_1$ can be related via a relation $R$ to its spatial extension in a spatial logic such as RCC8 (i.e., to a regular closed set of points in a topological space)—see Example 5; (3) two points $a$ and $b$ of a description logic can be related via a link relation has_orientation to two oriented points in a model of OPRA$_2$. Note that for this to work coherently, we need to make link relations functional, as can be achieved by adding number restrictions on links (see below).

Essentially, the language of an $\mathcal{E}$-connection is the (disjoint) union of the original languages enriched with operators capable of talking about the link relations.

The possibility of having multiple relations between domains is essential for the versatility of this framework, the expressiveness of which can be varied by allowing different language constructs to be applied to the connecting relations.

Figure 6 displays an example of the connection of two ontologies, with a single link relation $E$.

We first sketch the formal definitions for the 2-dimensional case. The reader is referred to [41] for involved examples and technical results on the computational properties of various specific $\mathcal{E}$-connections.

To formulate a 2-dimensional $\mathcal{E}$-connection between two ontologies $O_1$ and $O_2$ formulated e.g. in two different DLs DL$_1$ and DL$_2$ (here, an ontology is a set of axioms in the respective DL), we assume that
To form a connection of two ontologies by means of a single link relation \( E \), the **signatures** \( \mathcal{L}_1 = \text{Sig}(DL_1) \) and \( \mathcal{L}_2 = \text{Sig}(DL_2) \) of the two DLs, i.e., their sets of atomic concepts, roles, and object names, are pairwise disjoint.

To form a connection \( \mathcal{C}^E(DL_1, DL_2) \), fix a non-empty set \( \mathcal{E} = \{ E_j \mid j \in J \} \) of binary relation symbols. The **basic \( \mathcal{E} \)-connection**, then, has as signature the disjoint union of \( \mathcal{L}_1, \mathcal{L}_2 \) and \( \mathcal{E} \); its concept language is two-sorted with sorts \( s_1 \) and \( s_2 \) and defined by simultaneous induction as follows.

- (i) If \( C \) is a concept in \( DL_1 \), then \( C \) is of sort \( s_1 \); (ii) if \( D \) is of sort \( s_2 \) then \( \langle E_j \rangle^1 D \) is of sort \( s_1 \); (iii) \( s_1 \) is closed under the concept-forming operations of \( DL_1 \).
- (i) If \( D \) is a concept in \( DL_2 \), then \( D \) is of sort \( s_2 \); (ii) if \( C \) is of sort \( s_1 \) then \( \langle E_j \rangle^2 C \) is of sort \( s_2 \); (iii) \( s_2 \) is closed under the concept-forming operations of \( DL_2 \).

Here, the \( \mathcal{E} \)-connection-operators \( \langle E_j \rangle^1 \) and \( \langle E_j \rangle^2 \) are new concept-formation operators, interpreting the added link relations. The formal semantics is as follows: the class of models of \( \mathcal{C}^E(DL_1, DL_2) \) comprises all structures of the form

\[
\mathfrak{M} = \left( \mathfrak{M}_1, \mathfrak{M}_2, \mathfrak{E}^\mathfrak{M} = \{ E_j^\mathfrak{M} \}_{j \in J} \right),
\]

where \( \mathfrak{M}_i = (W_i, \mathfrak{M}_i) \) is an interpretation for \( DL_i \) for \( i \in \{ 1, 2 \} \) and \( E_j^\mathfrak{M} \subseteq W_1 \times W_2 \) for each \( j \in J \).

Given concepts \( C_i \) of ontology \( DL_i \), for \( i = 1, 2 \), denoting subsets of \( W_i \), the semantics of the basic \( \mathcal{E} \)-connection operators is

\[
\begin{align*}
(\langle E_j \rangle^1 C_2)^\mathfrak{M} &= \{ x \in W_1 \mid \exists y \in C_2^\mathfrak{M} (x, y) \in E_j^\mathfrak{M} \} \\
(\langle E_j \rangle^2 C_1)^\mathfrak{M} &= \{ x \in W_2 \mid \exists y \in C_1^\mathfrak{M} (x, y) \in E_j^\mathfrak{M} \}
\end{align*}
\]

Note that the requirement of disjoint domains is not essential for the expressivity of \( \mathcal{E} \)-connections. What is essential, however, is the disjointness of the formal languages of the component logics. What this boils down to is the following simple fact: while more expressive \( \mathcal{E} \)-connection languages allow to express various degrees of qualitative identity, for instance, by using number restrictions on links to establish partial bijections (which we will use below), they lack means to express 'proper' numerical trans-module identity.

It remains to clarify what the **sentences** of a basic \( \mathcal{E} \)-connection are. These just follow the same grammar as the component logics (in the case of DLs concept subsumptions, ABox and RBox statements), but respect the enriched concept language, with the obvious semantics interpreted in the local models. Moreover, we have ABox-like sentences for the link relations such as

\[
\mathfrak{M} \models (a, b) : E_j \iff E_j^\mathfrak{M}(a^\mathfrak{M}, b^\mathfrak{M}).
\]

Fig. 6 displays the connection of two ontologies by means of a single link relation \( E \). Here, the concept \( \langle E_j \rangle^1 \{ a \} \) of \( O_1 \) ‘corresponds’ to the nominal \( \{ a \} \) of ontology \( O_2 \): it collects the set of all those points in \( O_1 \) that ‘can be seen’ from \( a \) (in \( O_2 \)) along the relation \( E \).
Example 4 (Connecting two ontologies) Suppose two ontologies \( O_1 \) and \( O_2 \), formulated in different DLs \( DL_1 \) and \( DL_2 \), contain the concept \( \text{Window} \). Now, ontology \( O_1 \) might formalise functionalities of objects found in buildings, while ontology \( O_2 \) might be about the properties of materials of such objects. The intended relation between the two instances of \( \text{Window} \) might now be one of polysemy (meaning variation), i.e., \( \text{Window} \) in \( O_1 \) involves ‘something with views that can be open or closed’:

\[ \text{Window} \sqsubseteq \exists \text{has\_state} (\text{Open} \sqcup \text{Closed}) \sqcap \exists \text{offers\_Views}, \]

while the meaning of \( \text{Window} \) in \( O_2 \) might be ‘something that is bulletproof glass’:

\[ \text{Window} \equiv \text{Glass} \sqcap \exists \text{has\_feature.Bulletproof}. \]

A systematic integration of these two ontologies could now require a mapping of objects in \( O_1 \) to the material they are made from, using a link relation ‘\( \text{consists\_of} \)’. A concept of the form \( \langle \text{consists\_of} \rangle^1 \text{C} \) then collects all objects of \( O_1 \) that are made from something in \( C \), while a concept \( \langle \text{consists\_of} \rangle^2 \text{D} \) collects the materials in \( O_2 \) some object in \( D \) consists of. A sensible alignment between the two instances of \( \text{Window} \), introducing disjoint vocabulary \( \text{Window}_1 \) and \( \text{Window}_2 \), could now be formalised in \( \mathcal{E} \)-connections as:

\[ \langle \text{consists\_of} \rangle^2 \text{Window}_1 \sqsubseteq \exists \text{has\_feature.Transparent} \]
\[ \langle \text{consists\_of} \rangle^1 \text{Window}_2 \sqsubseteq \text{Window}_1 \sqcap \exists \text{provides\_security.Inhabitant} \]

assuming that windows in \( O_1 \) might also be made of plastic, etc.

Example 5 (Modelling Architectural Design) We sketch a first formal example on the use of \( \mathcal{E} \)-connections for modelling architectural design involving both conceptual and spatial dimensions: extending the previous example, let us suppose that we have a third dimension, a knowledge base formalised in the Region Connection Calculus RCC8 (as encoded in the modal logic \( S4_u \)) that we have introduced on Page 7. The following constraint is taken from [34] and illustrates the kind of modelling that can be performed in this setup.

[...] sensors have to cover certain regions around doors. These are functional regions that are defined by the doors and instantiated in the qualitative layer. The region of the sensor range has to be an inverse proper part of this functional region.

Here, ‘door’ and ‘sensor’ are taken as concepts \( \text{door} \), \( \text{sensor} \) that live in ontology \( O_1 \) introduced above. Moreover, we introduce two new relations bridging ontology \( O_1 \) and the RCC8 domain, namely \( \text{has\_functional\_space} \) that relates the instances of \( \text{door} \) with their functional space, i.e. regions in RCC8, and \( \text{has\_range\_space} \), again giving the regions covered by the sensors, see Fig. 7.

Here, models for an \( \mathcal{E} \)-connection of ontology \( O_1 \) and RCC8 are of the form

\[ M = \langle M_1, M_2, \mathcal{E}^{\mathcal{M}} = \langle \text{has\_range\_space}^{\mathcal{M}}, \text{has\_functional\_space}^{\mathcal{M}} \rangle \rangle, \]
where \( \mathcal{W}_1 \) interprets ontology \( O_1 \), \( \mathcal{W}_2 \) interprets RCC8, and the link relations are interpreted as subsets of the cartesian products of the domains of \( \mathcal{W}_1 \), \( \mathcal{W}_2 \). The constraint can now be formalised thus:

\[
PP^{-1}(\langle \text{has\_range\_space} \rangle^3 \text{sensor}, \langle \text{has\_functional\_space} \rangle^3 \text{door})
\]

Here, e.g. \( \langle \text{has\_range\_space} \rangle^3 \text{sensor} \) defines a region by collecting, for a given model \( \mathcal{M} \) of the \( \mathcal{E} \)-connection, all points in the RCC8 model that are ‘connected’ by the role \( \text{has\_range\_space} \) to an element of the concept \( \text{sensor} \).

To ensure that the regions thus obtained are ‘well-behaved’, we might want to enforce that they are regular-closed sets. This is a typical assumption in RCC8 based reasoning and can be formulated in \( \mathcal{E} \)-connections as follows:

\[
\langle \text{has\_range\_space} \rangle^3 \text{sensor} = CI \langle \text{has\_range\_space} \rangle^3 \text{sensor}
\]

using the closure operator \( C \) and interior operator \( I \) of \( S4_u \).

\( \mathcal{E} \)-connections can be considered as many-sorted heterogeneous theories: component ontologies can be formulated in different logics, but have to be built from a many-sorted vocabulary, and link relations are interpreted as relations connecting the sorts of the component logics. The main difference between distributed description logics (DDLs) [9] and various \( \mathcal{E} \)-connections now lies in the expressivity of the ‘link language’ \( L \) connecting the different ontologies. While the basic link language of DDL is a certain sub-Boolean fragment of many sorted \( \mathcal{ALC} \), the basic link language of \( \mathcal{E} \)-connections is \( \mathcal{ALC}^{Tms} \).

Such many-sorted theories can easily be represented in a diagram as shown in Fig. 8. Here, we first (conservatively) obtain a disjoint union \( S^m_1 \cup S^m_2 \) as a pushout (see [40] for technical details), where the component ontologies have been turned into sorted variants, and the empty interface guarantees that no symbols are shared at this point. An \( \mathcal{E} \)-connection knowledge base (KB) in language \( C^E(S^m_1, S^m_2) \) is then obtained as a (typically not conservative) theory extension.

Given that \( \mathcal{E} \)-connections have a relatively intuitive and simple semantics (compared e.g. to some multi-dimensional logics [21] or fibrings [20]), they remain quite popular as a modelling paradigm for heterogeneous combinations of ontologies with other formalisms. Apart from being applied in Semantic Web related research [26, 15], we applied them to model architectural design in previous work [6, 34], but they have also been employed e.g. to model heterogeneous combinations of linguistic ontologies and spatial

\[PP^{-1} \] here is the abbreviation for the union of \( TPP^{-1} \) and \( NTPP^{-1} \).

\[11\] But can of course be weakened to \( ALC^{Tms} \) or sub-Boolean DL, or indeed strengthened to more expressive many-sorted DLs involving e.g. number restrictions or Boolean operators on links, see [41] for details.
Fig. 8. $\mathcal{E}$-connections as a structured heterogeneous theory

calculi [31] which is also relevant for the application to design discussed in this paper. To give some more
detail on this latter usage of $\mathcal{E}$-connections, [31] analyse the problem of relating an ontology encoding
the linguistic spatial semantics of natural language utterances as represented in the linguistic ontology
GUM [4, 3] with spatial calculi, using the example of the double-cross calculus DCC [18] for projective
relations (orientations). The general relation between GUM and DCC is a loose coupling as can be ade-
quately modelled by an $\mathcal{E}$-connection. However, two entirely independent layers need to be added for a
‘complete’ formal representation of a spatial configuration: domain knowledge including naïve physics
information is added in a KB $\mathcal{D}$, while contextual information (such as intrinsic orientations, reference
system, etc.) is added by a KB $\mathcal{O}$. Both these layers of information are typically formalised in different
(heterogeneous) logics. The overall integration is obtained via a pushout operation, as shown in the upper
part of Fig. 8, taking $S_1 = \text{GUM}$ and $S_2 = \text{DCC}$. [30] take this kind of modelling approach a step further
by defining a variant of $\mathcal{E}$-connections, called $S$-connections, which introduces notions of similarity both
to the component logics as well as to the link relations, based on work on similarity [60] and distance
logics [38]. Here, ‘local similarity’ compares objects within one domain, whilst comparing objects across
domains leads to similarity measures that are motivated by and based on counterpart-theoretic semantics
[37].

6. Ontological Modelling of Form and Function: An Exploratory Study

Section 4.1 gave an overview of the different types of information that have to be combined when design-
ing an architectural environment and formulating architectural requirements. These different perspectives
comprise conceptual or domain-specific as well as spatial qualitative and quantitative aspects. Keeping
these different aspects separate in an ontological modelling framework not only reflects their actual dif-
ference but also helps in specifying form and function and their relationship in a more appropriate way.
Section 3 has introduced in detail various formalisms available for relating and combining different on-
tological specifications on a theoretical level. The main aim in our approach is now to (1) ontologically
specify the different perspectives on space necessary for architectural design specifically for the interplay
between form and function and (2) apply modular and heterogeneous specification formalisms in order
to reflect the different perspectives appropriately and to allow intelligible re-use and application of these
modular ontologies.

In the following, we introduce our ontological formalisation for the different spatial criteria that contribute
to an architectural design in general. We subsequently illustrate how architectural requirements about
form and function can be specified on the basis of these spatial modules. We subsequently present the
formalisation of particular building code requirements in more detail.

6.1. Modelling Heterogeneous Types of Spatial and Conceptual Information

Modules for architectural design can be thematically distinguished into conceptual, qualitative, and quan-
titative layers [34]. Each layer reflects closely related criteria about architectural design addressing either
conceptual, qualitative, or quantitative aspects.
Conceptual Aspects of Architectural Design. Modules in the conceptual layer (M1) reflect entities related to architectural design on the basis of their entity-based characteristics, i.e., they are specified by their properties and axioms without any contextual or embedded aspects. The layer can therefore extend existing foundational or general domain-specific entities. This can technically be done by importing and re-using (i.e., conservatively extending) the existing ontologies and refining their categories and relations if necessary. We developed a modular ontology in this layer that builds on and extends a foundational ontology, which provides an abstract foundation for specifying specific domain entities and relations, namely DOLCE [47]. In particular, its OWL version DOLCE-Lite\textsuperscript{12}, is refined in order to provide a categorisation of architectural entities. The resulting ontology refines physical-object and non-physical-endurant of DOLCE. It introduces physical entities, such as Staircase, Desk, SlidingDoor, and PhysicalRoom, and it introduces functional non-physical endurants, in particular different building types, such as ApartmentBuilding, OfficeBuilding, and University, and different room types, such as Lobby, Office, Salesroom, and Bathroom. These functional aspects are related to a PhysicalRoom in the following way (omitting the inferred constraints from inherited categories in DOLCE):

\[
\text{BuildingType} \sqsubseteq \text{dolceLight:non-physical-endurant} \\
\sqsubseteq \exists \text{dolceLight:generically-dependent-on} . \text{PhysicalBuilding}
\]

This constraint defines that types of building, e.g., university, church, museum, are non-physical endurants and they depend on an actual (physical) building which provides this building type or function. This formalisation is inspired by the modelling of artefacts and roles introduced in [66]. The namespace ‘dolceLight’ in the formula indicates which parts are re-used from the foundational ontology. Parts without a namespace are specified in the conceptual module that refines DOLCE. While the categories specified in the conceptual layer can be related to an intended architectural design, conceptual aspects could also include more abstract types of information, e.g., about costs, environment, user groups, or actions. This information is then specified by another module in the conceptual layer.

Qualitative Aspects of Architectural Design. Modules in the qualitative layer (M2) specify entities related to architectural design on the basis of their qualitative spatial characteristics. Each module in this layer can thus describe spatial entities with regard to topology, distance, orientation, or other qualitative spatial aspects (discussed in Section 3). In particular, spatial logics or calculi, which are introduced in Section 3.2, are part of this layer. Information about qualitative spatial relationships are useful not only to define basic relationships between entities of an architectural design, e.g., doors are connected with walls or windows, but also to define their functional requirements, e.g., staircase landings are not supposed to overlap with functional regions of doors. Analysing whether an architectural design satisfies given requirements can thus be achieved by defining restrictions on qualitative spatial models of an architectural design. The staircase landing example can thus be formalized by requiring the regional extension of the landing to be (RCC-)disconnected with the regional extension of the functional areas of doors (cf. formulas in Section 6.2). This modelling then uses $\mathcal{E}$-connections as defined above, in order to formalize constraints between modules from the conceptual and qualitative layer.

The basic requirement that doors in an architectural design are necessarily connected with either walls, windows, or doors, is formalized by constraints of the spatial extension of these concepts. This requirement is related to region-based information, and thus the regional extension of any door needs to be externally connected with walls, windows, or doors, with regard to the RCC relations. The following formula defines this particular requirement in the $\mathcal{E}$-connected theory of a module from the conceptual and the qualitative layer. For brevity, we define the following regions in the qualitative module:

\[
\begin{align*}
\text{OpSpaceDoor} & = \langle \text{has\_operational\_space} \rangle^{M2} \text{Door} \\
\text{OpSpaceWall} & = \langle \text{has\_operational\_space} \rangle^{M2} \text{Wall} \\
\text{OpSpaceWindow} & = \langle \text{has\_operational\_space} \rangle^{M2} \text{Window}
\end{align*}
\]

\textsuperscript{12}DOLCE-Lite: \url{http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl}
The index $M^2$ indicates the qualitative module (cf. Fig. 9). The following constraint now encodes the requirement above:

$$\text{EC}(\text{OpSpaceDoor, OpSpaceDoor} \sqcup \text{OpSpaceWall} \sqcup \text{OpSpaceWindow})$$

Such constraints formulate basic requirements an architectural design has to satisfy in general. They primarily reflect the kinds of spatial relations given by a spatial calculus, which can again be based on regions, orientations, distances, or shapes. As seen above, legal issues and building codes require even more abstract and complex constraints on a particular design. While the basic constraints ensure that a building complies with primitive spatial standards, the complex constraints ensure that a building conforms to particular design standards. Therefore, primitive kinds of constraints are formalized for the $\mathcal{E}$-connected conceptual and qualitative layer, complex kinds of constraints are formalized in an additional layer for particular building code requirements (cf. Section 6.2).

Proving whether an architectural design satisfies a set of given requirements, which are formulated in a logical module or an $\mathcal{E}$-connected theory, is based on ontological reasoning, in particular ABox reasoning. As long as no constraints are violated, the architectural design (instantiate as an ABox) complies with the given requirements.

**Quantitative Aspects of Architectural Design.** Modules in the quantitative layer ($M^3$) specify entities related to architectural design on the basis of metrical information. They particularly reflect metrical data of construction elements in building plans, e.g., heights of ceilings, positions of walls, widths and heights of windows, or opening angles of doors. Modules in this layer are closely related to standards and tools for architectural design.

**Industry Foundation Classes:** In previous work [6, 34], we apply the Industry Foundation Classes (IFC) [19], which is a data model for architectural design and design tools. It aims at interoperability in the building industry by providing a non-proprietary data exchange format that reflects constructional information about buildings. IFC can be related to 3D CAD models, though its data model is more expressive than CAD. IFC not only defines geometric primitives, such as points, lines, and polygons, and raw metrical data about these entities, it also defines primitive semantics for them by relating these objects to structural elements: it defines concrete building components like walls, windows, or roofs, as well as abstract entities like actions, spaces, or costs. As the data format is supported by commercial as well as free software design tools, which also allow exports into other XML and binary formats, for modelling, visualising, or syntax checking, its applicability is guaranteed. In particular, as we use the IFC in our approach, datasets from other IFC compliant design tool can easily be used as well. IFC specifies different types of building entities that provide a basis for an ontology module in the quantitative layer. This module resembles relevant IFC classes necessary for formulating functional requirements of a design. IFC provides the different architectural entities of a design and their basic properties. For instance, a door in IFC is defined as IfcDoor that has (i) a door width and height, and (ii) a door opening direction (with regard to the y-axis). This information is accordingly encoded in a description logic ontology as part of the quantitative layer, as follows:

$$
\text{Door} \sqsubseteq \text{StructuralBuildingElement} \sqsubseteq 1 \text{openingAngle} \cdot \text{float} \\
\sqsubseteq 1 \text{railingType} \cdot \text{ENTITY} \sqsubseteq 1 \text{height} \cdot \text{float} \\
\sqsubseteq 1 \text{length} \cdot \text{float} \sqsubseteq 1 \text{width} \cdot \text{float}
$$

Modules in this quantitative layer are primarily related and linked to the qualitative layer. It can metrically ground the qualitative spatial relations between entities. Furthermore, the entities defined in the quantitative layer also relate to information in the conceptual layer, e.g., the specific types of doors ‘revolving door’, ‘sliding door’, or ‘swing door’.
Connecting Modular Ontologies from Different Layers for Architectural Design. Having introduced the three main aspects about architectural design that can be distinguished spatially and ontologically, we formulate their connections and define primitive architectural requirements. As all three layers refer to the same entity, namely an architectural design, they are all related with each other. Conceptual aspects are relevant for the entities in the quantitative layer. Qualitative aspects are relevant for the spatial relationships between entities from both conceptual and quantitative layers. Hence, the connection between the different components result in a three-dimensional $E$-Connection, as illustrated in the example in Fig. 9.

![Fig. 9. A 3-Dimensional $E$-connection, relating the conceptual, qualitative, and quantitative modules.](image)

The figure shows an example of a part of an architectural model of a design. Here, the basic relationships between entities and their connections across layers are defined. In the conceptual module, a particular type of a door (a swing door) is related to an entrance point that refers to the place, which can be entered by the door. In the example, the swing door is the entrance to a kitchen. The qualitative region that spatially extends the swing door is defined in the qualitative module. This entity is related by an $E$-Connection between the modules. In particular, as introduced above, the link relation $\text{has\_operational\_space}$ connects the swing door with a region that reflects its operational space while the link relation $\text{has\_functional\_space}$ connects the door with a region that reflects its functional space. Indices in Fig. 9 indicate which modules are related by these link relations. Within the qualitative module, regions can spatially be related with each other. For instance, the operational space of the door is a proper part of the functional space of the door and it is disjoint with another region that is related to a column. In the example, the column is given by a quantitative model of the architectural design, which is linked from the quantitative model to the qualitative module. The quantitatively specified column is based on its metrical information, e.g., its diameter. Its position is also located in a room in the architectural floor plan. This room is connected with the conceptual kitchen by an $E$-Connection between the quantitative and conceptual layer. The link relation $F$ between conceptual and quantitative modules are specified as being (logically) functional (cf. Table 2), as each concept only refers to exactly one metrical entity, and similarly, each metrical entity (a polygon, line, etc.) in the floor plan refers to only one conceptual entity (a specific wall, window, etc.).

The $E$-Connections $E, F, G$ in Fig. 9 between the different layers can thus be used to link the various types of information about the same architectural design. In general, certain types of elements from each of the three layers have to be $E$-connected with elements from other layers. While the example in Fig 9 introduces link relations for specifying basic requirements between the three modules, the next section presents specific functional requirements as required by particular building code standards.
6.2. Modelling Architectural Design and Functional Requirements with Spatial Ontologies

Architectural design requirements are usually given by a set of (natural language) descriptions that also include functional requirements. Within our modelling of architectural design, these sets form a module in a requirement layer that builds on the three-level formalisation for conceptual, qualitative, and quantitative information. They enhance and refine the requirements among components that are more specific and complex than the primitive building constraints. They therefore technically extend the ontological $\mathcal{E}$-Connection theory introduced above, as illustrated in Fig. 10. The extension defines particular requirements or constraints that have to be satisfied by an architectural design, e.g., in order to fulfil legal requirements or building codes.

![Diagram](https://example.com/diagram.png)

**Fig. 10.** Functional requirements of a particular architectural design (e.g., a building code) can be defined by (conservatively) extending the 3-dimensional formalisation for conceptual, qualitative, and quantitative space. Each layer can consist of several logical theories or modular ontologies, e.g., the qualitative layer consists of spatial logics and calculi for regions, orientations, and distances.

The German building code referred to in Section 2 is one example of such a requirement module that specifies building functionalities and constraints, such as the landing example. Repeating one of its official regulations:

“Steps of a staircase may not be connected directly to a door that opens in the direction of the steps. There has to be a landing between the staircase steps and the door. The length of this landing has to have at least the size of the door width.” (Bremen (Germany) Building code [10])

Given this (natural language) constraint, its interpretation in terms of conceptual, qualitative, and quantitative spatial categories has to be specified in the related requirement module for the building code. It is primarily reflected by region-based constraints of the entities in the qualitative module. These entities are related to their qualitative counterparts describing staircases and their landings in the quantitative module, i.e., in the actual floor plan. These entities may also be connected with conceptual elements that reflect particular aspects of the staircase, such as material or tread types.

The landing requirement is specified by the ontological requirement that all operational spaces of staircase landings may not overlap with operational spaces of doors. The requirements, while formalized in the requirement module, constrains the link relation between the quantitative and the qualitative module:

\[
\text{OpSpaceStaircaseLanding} = (\text{has\_operational\_space})^{M_1} (\text{Landing} \sqcap \exists \text{landingOf}\text{.Staircase})
\]

\[
\text{OpSpaceDoor} = (\text{has\_operational\_space})^{M_1} \text{Door}
\]
Again for brevity, we introduce the regions `OpSpaceStaircaseLanding` (the operational space of a landing of a staircase) and `OpSpaceDoor` (the operational space of a door), both in the qualitative module. The index `M1` indicates that the categories `Landing`, `Staircase`, and `Door` are defined in the quantitative module. Necessarily these regions have to be disjoint with each other, which reflects the building code requirement. A given architectural design can thus be proven to satisfy this requirement by its ABox consistency analysis. In this manner, each regulations from the building code can be specified in the requirement module, and a design that satisfy these requirements consequently satisfies the building code.

While the landing example basically constrains structural building aspects of a design, requirements can be more complex, such as the functional requirements given by the courthouse design guide introduced in Section 2. We repeat the requirement here for convenience:

> “Witnesses must be able to see and hear, and be seen and heard by, all court participants as close to full face as possible. The witness box must accommodate one witness and an interpreter, and the preferences of the presiding judge. Witnesses in the box receive, examine, and return exhibits.” (US Courts Design Guide 2007 [65])

This requirement contains conceptual information about witnesses, interpreters, judges, and participants as particular persons or user groups and about receiving, examining, and returning exhibits, seeing and hearing as specific actions. Spatial constraints require that certain persons are supposed to be in close distance with each other, that they are supposed to face each other, and that the witness box provides enough space for certain persons. Hence qualitative spatial relations about distance, orientations, and size or region are postulated.

The requirement is specified in the requirement module for Courthouse Design Guides as follows. The spatial calculus `OPRA2` [49] is used for modelling orientations. The index `O1` in the formulas encode the qualitative module for this orientation calculus. Participants and witnesses are here required to face each other at least within 90° to both sides. The spatial distance logic [38] is indexed with `D1`:

1. “Witnesses must be able to see and hear, and be seen and heard by, all court participants as close to full face as possible”

   \[
   \begin{align*}
   \text{FrontParticipant} & = \langle \text{has\_orientation} \rangle^{O1} (\text{Participant}) \\
   \text{FrontWitness} & = \langle \text{has\_orientation} \rangle^{O1} (\text{Witness}) \\
   \end{align*}
   \]

   \[
   \begin{align*}
   \text{OPrA}_{2} \angle_{0}^{0} (\text{FrontWitness}, \text{FrontParticipant}) \sqcup \text{OPrA}_{2} \angle_{1}^{0} (\text{FrontWitness}, \text{FrontParticipant}) \\
   \sqcup \text{OPrA}_{2} \angle_{0}^{1} (\text{FrontWitness}, \text{FrontParticipant}) \sqcup \text{OPrA}_{2} \angle_{0}^{0} (\text{FrontWitness}, \text{FrontParticipant}) \\
   \sqcup \text{OPrA}_{1} \angle_{1}^{1} (\text{FrontWitness}, \text{FrontParticipant}) \sqcup \text{OPrA}_{1} \angle_{1}^{0} (\text{FrontWitness}, \text{FrontParticipant}) \\
   \sqcup \text{OPrA}_{0} \angle_{1}^{0} (\text{FrontWitness}, \text{FrontParticipant}) \\
   \end{align*}
   \]

   \[
   \begin{align*}
   \text{PosWitness} & = \langle \text{has\_distance\_space} \rangle^{D1} (\text{Witness}) \\
   \text{PosParticipant} & = \langle \text{has\_distance\_space} \rangle^{D1} (\text{Participant}) \\
   \text{pd} & = \langle \text{has\_distance} \rangle^{D1} \text{privacyDistanceValue} \\
   \text{cd} & = \langle \text{has\_distance} \rangle^{D1} \text{communicationDistanceValue} \\
   \text{PosWitness} \rightarrow A_{<pd}^{>pd} \text{PosParticipant} \\
   \text{PosWitness} \rightarrow A_{<cd}^{>cd} \neg \text{PosParticipant}
   \end{align*}
   \]

The spatial calculus RCC8, given by the index `R1`, is used for modelling regions. Witnesses and Interpreters are supposed to be placed inside the witness box:
An illustration of the spatial constraints in the different qualitative modules is given in Fig. 11. In this way, functional and informally defined design requirements can be formally specified by using the three-dimensional representation of the different spatial aspects. If an architectural design satisfies the ontological requirements, it also satisfies the design guide.

7. Discussion and Outlook

We have focussed on the formalisation of design semantics, primarily encompassing aspects pertaining to the structural form of a spatial design, and a high-level specification of design requirements occurring thereof. The modelling approach aims to make the representation accessible for automated reasoning capabilities concerned with providing an analytical function during the initial design conceptualisation and iterative refinement phase, as identified in the context of spatial computing for design [7]. Here, spatial computing (for design) is defined as:

- “the body of work that is concerned with the use of formal methods in knowledge representation and reasoning in general, and terminological and spatial representation and reasoning in specific, for solving problems in modelling (e.g., spatial semantics, modularity, requirement constraints) and validation (e.g., diagnosis, hypothetical reasoning) in the domain of spatial design”
- “that body of work whose aim is to develop the generic apparatus— application framework, methodology, tool-sets —that may be used as a basis of providing assistive design support within a conventional CAAD-based spatial design workflow”

Situated within this AI-centred view of spatial computing for design, this paper has addressed the need to formally represent and reason about (structural) form & (artefactual) function. Specifically, the paper focussed on semantic modelling, spatial abstraction, and multi-perspective representation. These aspects, as considered in this paper, accrue within a conventional ‘iterative refinement by automated design assistance’ workflow, and are identifiable with respect to the modelling–evaluation–re-design phases in intelligent design assistance, for instance, as interpreted within the ontological framework of the Function-Behaviour-Structure (FBS) model [22, 23] of the design process. An overview of the envisaged design workflow is illustrated in Fig. 12: first, a work-in-progress design is modelled in an architectural design tool. Then, for every module, the data model formulated in a CAAD model is instantiated as an ABox. Given certain task-specific or functional requirements for the environment being modelled, spatio-
terminological reasoning supported by different reasoning components proves (or disproves) the consistency of the work-in-progress design. The results are then incorporated within the iterative refinement phase, repeating this process until certain design objectives are satisfied, i.e., no requirement inconsistencies occur. Indeed, this description of this work-flow and the design assistance therein is rather limited; an in-depth discussion is available in [7]. With respect to the refinement work-flow, the basic research questions, in so far as the scope of this paper is concerned, within the context of spatial computing include:

1. **Semantics**: formal modelling of design requirements, and the role of knowledge engineering in general, and ontological engineering in particular
2. **Spatial abstraction**: abstraction of CAD-based geometric information into the qualitative domain via the use of formal spatial representation and reasoning techniques

In this paper, we have concentrated on the complex modelling problems that are encountered in the domain of architectural design. In order to account the need to incorporate heterogeneous multi-perspective representations in design, we have adopted the formal framework of $\mathcal{E}$-connections. The formal framework of $\mathcal{E}$-connections not only facilitates representational modularity, but also supports automatic reasoning rather well by ensuring the decidability of the global reasoning problems with diverse logical reasoning modes, such as reasoning over topological relations, distances, etc. As discussed in Section 4, modularity in ontological design has many facets. In [52], for instance, modularity is motivated by the complexities of designing large biomedical ontologies: they utilise normalisation—as in a database sense—as the underlying approach to achieve modularity for arbitrary DL ontologies, including OWL ontologies. Such an approach to modularity could potentially also serve as a basis of modularisation in the architectural domain if the ontological knowledge involved was expected to undergo continuous evolution, and distributed design and development. In our case, however, the taxonomic knowledge is more or less static since the IFC is a de facto design standard and not subject to regular modifications. Nevertheless, comparing the different approaches to modularity and their impact on the reasoning capabilities is an important topic for further consideration. Non-classical inference, of course, is of particular interest in a domain such as architectural design, in particular hypothetical and para-consistent inference, see e.g. [63, 7]. Reasoning support for $\mathcal{E}$-connections combining e.g. DLs with RCC8 is under active development, and the extension to non-classical reasoning problems will remain as a very promising direction for future research.

In comparison with existing upper level ontologies that describe functions, for instance, introduced in [35, 48], the type of function we present in this paper is reflected closely by the concept of *requirement function* in those ontologies. Here, “function is a role played by a behavior specified in a context” [48]. In the context of architectural design, however, we adopt a notion of functions that is strongly interpreted in terms of its possible qualitative and quantitative spatial models formalized by using different perspectives on space. Hence, no upper-level ontology of functions was re-used explicitly in the conceptual modelling of architectural design. For future work, however, ontologically axiomatising the relationship between abstract or loosely defined requirements for architectural design and their modular formalisation across spatial perspectives is also an important future direction.
Acknowledgements

We acknowledge the financial support of the DFG through the Collaborative Research Center SFB/TR 8 Spatial Cognition. Participating projects in this paper include: [DESIGNSPACE] and 11-[ONTOSPACE]. The first author also acknowledges funding by the Alexander von Humboldt Stiftung, Germany. We thank Graphisoft – http://www.graphisoft.com/ – for providing licenses for the design tool ArchiCAD v13 2010. We thank John Bateman for constructive discussions and feedback.

References


