Spatio-Terminological Inference for the Design of Ambient Environments

Mehul Bhatt, Frank Dylla, and Joana Hois

SFB/TR 8 Spatial Cognition
University of Bremen, Germany

Abstract. We present an approach to assist the smart environment *design process* by means of automated validation of work-in-progress designs. The approach facilitates validation of not only the purely structural requirements, but also the functional requirements expected of a smart environment whilst keeping in mind the plethora of sensory and interactive devices embedded within such an environment. The approach, founded in spatio-terminological reasoning, is illustrated in the context of formal ontology modeling constructs and reasoners, industrial architecture data standards and state-of-the-art commercial design software.

Keywords: Spatio-Terminological Reasoning, Ontology, Requirements Modeling, Smart Environment Design, Architecture.

1 Motivation

The field of Ambient Intelligence (AmI) is beginning to manifest itself in everyday application scenarios in public and private spheres. Key domains include security and surveillance applications and other utilitarian purposes in smart homes and office environments, ambient assisted living, and so forth [3, 30]. Notwithstanding the primarily commercial motivations in the field, there has also been active academic (co)engagement and, more importantly, an effort to utilize mainstream artificial intelligence tools and techniques as a foundational basis within the field [4, 26]. For instance, the use of quantitative techniques for sensor data analysis and mining, e.g., to look for patterns in motion-data, and for activity and behavior recognition has found wide acceptability [24, 33].

Shift in Design Perspective As AmI ventures start to become mainstream and economically viable for a larger consumer base, it is expected that AmI projects involving the design and implementation of smart environments such as smart homes and offices will adopt a radically different approach involving the use of formal knowledge representation and reasoning techniques [4, 6]. It is envisioned that a smart environment will be designed from the initial stages itself in a manner so as to aid and complement the requirements that would characterize its anticipated functional or intelligent behavior [1]. Presently, a crucial element that is missing in smart environment (and architecture) design pertains to the formal modeling – representation and reasoning – of spatial structures and artifacts contained therein. Indeed, since AmI systems primarily pertain to a spatial environment, formal representation and reasoning along the ontological (i.e., semantic make-up of the space) and spatial (i.e., configurations and constraints) dimensions can be a useful way to ensure that the designed model satisfies key functional requirements that enable and facilitate its required *smartness*. Broadly, it is this design approach in the initial modeling phase that we

operationalize in this research. Although the presented methods can be applied to general architecture as well, they are of specific interest in ambient environments as the number of entities is much higher and thus, keeping track of possible dependencies is more complex and complicated.

Absence of Semantics Professional architecture design tools are primarily concerned with the ability to develop models of spatial structures at different levels of granularity, e.g., ranging from low-fidelity planar layouts to complex high-resolution $3D$ models that accurately reflect the end-product. For instance, using a CAD tool to design a floor plan for an office, one may model various spatial elements representative of doors, windows, rooms, etc., from primitive geometric entities that collectively reflect the desired configuration. However, such an approach using contemporary design tools lacks the capability to incorporate and utilize the semantic content associated with the structural elements that collectively characterize the model. Furthermore, and partly as a consequence, these tools also lack the ability to exploit the expertise that a designer is equipped with, but unable to communicate to the design tool explicitly in a manner consistent with its inherent human-centered conceptualization, i.e., semantically and qualitatively. Our approach utilizes formal knowledge representation constructs to incorporate semantics at different layers: namely the conceptual or mental space of the designer and a quality space with qualitative abstractions for the representation of quantitatively modeled design data.

Semantic Requirements Constraints As a result of *absence of semantics*, it is not possible to formulate spatial (and non-spatial) requirement constraints, expressed semantically at the conceptual level, that may be validated against a work-in-progress design (i.e., the realization) at the precise geometric level. For instance, from a purely structural viewpoint, a typical requirement in an arbitrary architectural design scenario would be that the extensions of two rooms need to be in a particular spatial (topological or positional) relationship with each other – it may be stipulated that certain structural elements within a real design that are semantically instances of concepts such as ChemicalLaboratory and Kitchen may not be *next* to each other, or should be separated by a minimum distance. Such spatial constraints are important not because of the level of their inherent complexity from a design viewpoint, which is not too much, but rather because they are semantically specifiable, extensive, and hard to handle for a team of engineers collaboratively designing a large-scale, inter-dependent environment. Our approach formalizes the conceptualization and representation of such constraints in the context of practical state-of-the-art design tools.

1.1 Requirements Constraints in AmI Design

Semantic descriptions of requirement constraints acquires real significance when the spatial constraints are among strictly spatial entities as well as abstract *spatial artifacts*. This is because although *spatial artifacts* may not be spatially extended, but they need to be treated in a real physical sense nevertheless, at least in so far as their relationships with other entities are concerned. Since a conventional working design may only explicitly include purely physical entities, it becomes impossible for a designer to model constraints involving spatial artifacts at the design level, thereby necessitating their specification at a semantic level. For example, in the design of ambient intelligence environments, which is

the focus of this paper, it is typical to encounter and model relationships between spatial artifacts (see Section 3.2) such as in (A1–A3):

A1. the *operational space* denotes the region of space that an object requires to perform its intrinsic function that characterizes its utility or purpose
A2. the *functional space* of an object denotes the region of space within which an agent must be located to manipulate or physically interact with a given object
A3. the *range space* denotes the region of space that lies within the scope of a sensory device such as a motion or temperature sensor

Indeed, the characterizations in (A1–A3) are one set of examples relevant for the example scenario presented in this paper. However, from an ontological viewpoint, the range of potential domain-specific characterizations is possibly extensive, if not infinite. Constraints such as in (C1–C3) may potentially need to be satisfied with the limited set of distinctions in (A1–A3):

C1. the functional space of the door of every office should overlap with the range space of one or more motion sensors
C2. there should be no region of space on the floor that does not overlap with the range space of at least one camera
C3. key monitored areas that are connected by doors and/or passages should not have any security blind spots whilst people transition from one room to another

Constraints such as (C1–C3) involve semantic characterizations and spatial relationships among strictly spatial entities as well as other spatial artifacts. Furthermore, albeit being modeled qualitatively at a conceptual level, they also need to be validated against a quantitatively modeled work-in-progress design (e.g., a CAD model) in addition to checking for the consistency of a designer's requirements per se (Section 3.6).

1.2 Key Contribution and Organization

We apply the paradigm of integrated spatio-terminological reasoning for the design and automated validation of smart spaces. The validation encompasses the structural as well as functional requirements expected of a smart environment from the viewpoint of the sensory and interactive devices embedded within such an environment. In essence, a quantity space, modeled accurately using primitive geometric elements, is validated against a domain conceptualization consisting of ontology of spatial entities, artifacts, architectural elements, sensory devices and the relationships among these diverse elements.

Section 2 presents the ontological underpinnings of this work. Here, the concept of integrated spatio-terminological inference is illustrated and the use of spatial ontologies for AmI systems modeling is explained. Section 3 sets up the apparatus for formal requirement constraints modeling with ontologies. This is in turn utilized in the example scenario of Section 3.6, where the proposed approach is demonstrated in the context of an industrial standard for data representation and interchange in the architectural domain, namely the Industry Foundation Classes (IFC) [9], and state-of-the-art commercial architecture design techniques, as enabled by the ArchiCAD [7] design tool. Finally, Section 4 discusses the work in its relationship to existing research, whereas Section 5 concludes with pointers to the outlook of this research.[1]

[1] Additional (independent) information in support of the paper is linked at the end of the article.

2 Ontology, Architecture and Ambient Intelligence

Ontologies are defined as "a shared understanding of some domain of interest" [31]. Their structure consists of classes, relations between classes, and axiomatizations of classes and relations (cf. [28] for further information). In the case of AmI systems ontologies can then provide a formalization of entities, relations, and axiomatizations specifically for the AmI environment, as described below.

2.1 Ontologies and Spatio-Terminological Reasoning

Although ontologies can be defined in any logic, we focus here on ontologies as theories formulated in description logic (DL), supported by the web ontology language OWL DL [23]. In general, DL distinguishes between TBox and ABox. The TBox specifies all classes and relations, while the ABox specifies all instantiations of them. Even though ontologies may be formulated in more or less expressive logics, DL ontologies provide constructions that are general enough for specifying complex ontologies [17]. Several reasoners are available for DL ontologies, one of them is the reasoning engine RacerPro [13] with its query language nRQL. Here, we use this reasoner for spatio-terminological inference.

Reasoning over the TBox allows, for instance, to check the consistency of the ontology and to determine additional constraints or axioms that are not directly specified in the ontology. Reasoning over the ABox allows, for instance, to classify instances or to determine additional relations among instances. In particular for AmI ontologies, the domain of buildings and their ambient characteristics and constraints have to be specified in order to formalize their requirements. In addition to reasoning over ontologies (TBox) and their instances (ABox), however, spatial reasoning is of particular interest for the AmI domain, as spatial positions of entities and qualitative spatial relationships between entities are highly important to describe the environment.

A specific feature of the reasoning engine RacerPro is to support region-based spatial reasoning by the so-called SBox [11]. Besides TBox and ABox, this layer provides spatial representation and reasoning based on the Region Connection Calculus (RCC) [27]. The SBox can mirror instances of the ABox and specify RCC-relations and consistency among these instances. The separation between spatial and terminological representation and reasoning supports practicability of reasoning, reduces complexity, and benefits ontological modeling in general, as a domain can be described from different perspectives [25], e.g., with a focus on terminological, spatial, temporal, functional, action-oriented, or other thematically different perspectives. The integration of perspectives then allows a comprehensive representation of the domain. While each ontology specifies and axiomatizes its respective view on the domain, alignments across different ontologies provide further axiomatizations [19]. Our particular AmI ontological representation and reasoning are described in Section 3.5 and Section 3.6.

2.2 Industry Foundation Classes and Design Tools

Industry Foundation Classes (IFC) [21] are specific data models to foster interoperability in the building industry, i.e., a non-proprietary data exchange format reflecting building information. Former models, like 2D or 3D CAD models are based on metric data referring to geometric primitives, e.g., points, lines, etc., without any semantics of these primitives. In contrast, IFC is based on object

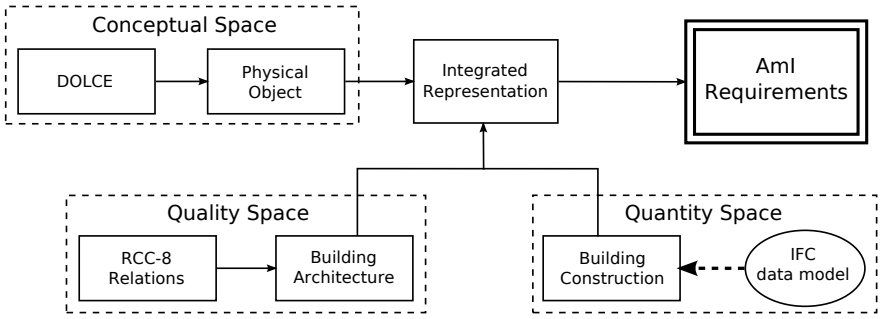


Fig. 1: Spatial ontologies for AmI: three ontological modules for conceptual, quality, and quantity space reflect the different perspectives on the domain.

classes, e.g., *IfcWall* or *IfcWindow*, and their inherent relationships containing metrical data as properties. The advantage of this kind of representation is that data for complex calculations like structural analysis or energy effort can be generated automatically. Within our work we apply the latest stable release IFC2x3 TC1 [21]. Overall, IFC 2x3 defines 653 *building entities* (e.g., *IfcWall*) and additionally, several *defined types*, *enumerations*, and *select types* for specifying their properties and relationships. Commercial design tools such as Graphisoft's ArchiCad [7] support export capabilities in XML and binary format in a manner that is IFC compliant. Free software tools also exist for modeling, visualizing, syntax checking, etc., of XML and binary IFC data. Note that since our approach utilizes IFC data, datasets from any IFC compliant design tool remain utilizable.

3 Requirements Consistency in AMI Design

In this section, the modular specification of ontologies for AmI that support architectural design processes are presented. We show how the architectural representation is associated with the ontological representation, how the IFC representation can be instantiated, and how ontological information can be grounded in architectural designs. Subsequently spatial and terminological requirements formulated within and across the ontologies are presented. An example illustrating reasoning and consistency checking is provided in Section 3.6.

3.1 Modular Spatio-Terminological Ontologies for AmI Design

As outlined in Section 2.1, space can be seen from different perspectives and ontologies are a method to formalize these perspectives. In order to support the architectural design process for AmI environments, requirement constraints of architectural entities have to be defined from terminological and spatial perspectives. Space is then defined from a *conceptual*, *qualitative*, and *quantitative* perspective. The resulting three modules consist of different ontologies, illustrated in Fig. 1.

M1. Conceptual Space This ontological module reflects terminological information of architectural entities. Here, the entities are regarded as such, i.e., they are defined according to their properties without taking into account the context in which they are put. The ontology Physical Object formalizes entities

with respect to their attributes, dependencies, functional characteristics, etc. It is based on DOLCE [22], in particular, on the OWL version DOLCE-Lite, and refines DOLCE's Physical Endurants. It defines the entities that occur in the AmI domain, such as Sensor, SlidingDoor, or ChemicalLaboratory.

M2. Quality Space This ontological module reflects qualitative spatial information of architectural entities. Similar to the previous module, the ontology Building Architecture formalizes entities of the AmI domain, but it specifies their region-based spatial characteristics. In particular, the ontology uses relations as provided by the spatial calculus RCC [27]. A Room, for instance, is necessarily a *proper part* of a Building. Here, we reuse an RCC ontology, that has been introduced in [10], which defines the taxonomy for RCC-8 relations by approximations of the full composition table. RCC-related constraints by the TBox can be directly inferred. Inference given by the composition table for combinations of RCC relations is then provided by SBox reasoning. Architecture-specific entities in the Building Architecture ontology are further described in Section 3.2.

M3. Quantity Space This ontological module reflects metrical and geometric information of architectural entities. It is closely related to the IFC data model and partially mirrors the IFC classes. In particular, the ontology Building Structure of the module specifies those entities of the architectural domain that are necessary to describe structural aspects of ambient environments. Especially, information that is available by construction plans of buildings are described here. For example, Door, Wall, and Window are characterized together with their properties length, orientation, placement, etc. Data provided by IFC for a concrete building model can then be instantiated with respect to the Building Structure ontology (cf. Section 3.3). Even though the IFC model itself is not an ontology, parts of it are directly given by their correspondences in the ontological specification in this module.

Integrated Representation The connection of the three different modules result in formalizing relations across modules. The Integrated Representation defines couplings between classes from different modules, i.e., counterparts and dependencies are defined across modules, based on the theory of $\mathcal{E}$-connections [20]. For example, an instance of Wall from the quantity space is related to an instance of Wall from the quality space or conceptual space. An IFC wall that is illustrated in a construction plan can then be instantiated as a Wall in Building Construction and connected to an instance of Wall in Building Architecture as well as an instance of Wall in Physical Object. It is described by its length and position in the first module, while its counterpart in the second module defines region-based relations to other walls and relations to rooms it constitutes and its counterpart in the third module defines its material and color. Details on modularly specified ontological modules for architectural design are given in [15].

3.2 Space: Objects and Artefacts

We present an informal characterization of the primitive spatial entities within the spatial ontology, or precisely, the modular component as reflected by the *quality space* within the overall spatial ontology (see Section 3.1; Fig. 1). For all ontological characterizations here, precise geometric interpretations are provided in Section 3.3. Here, a high-level overview suffices.

Regions are either the absolute spatial extensions of physical objects, or of spatial artifacts that are not truly physical, but are required to be regarded as such. A *region* of space should be measurable in terms of its area (2D) or volume (3D) and the region space should be of uniform dimensionality, i.e., it is not possible to express a topological relationship between a 2D and and 3D region. The spatial categories in (S1–S4) are identifiable. Spatial relationships between these categories are utilized for modeling structural and functional requirement constraints for a work-in-progress design:

S1. Object Space The *object space* of a primitive entity refers to the region covered by the physical extent of the respective entity itself. If objects are static, non-deformable, and reconfigurable they cover a well-defined region in the world. In contrast, if an entity is non-static, deformable, or reconfigurable, its spatial extension depends on its specific state s, e.g., the opening angle of a door or window. Let $obspace(o)$ and $obspace(o, s)$ denote the state independent and dependent space covered by such an object. Here, $\mathcal{S}_o$ is the set of all potential states an object may be in and $s \in \mathcal{S}_o$. Since we only deal with static worlds in our modeling, we abstract away from the state parameter s and simply use $obspace(o)$ to denote the space covered by the object in a specific *predefined state* s_p. We assume that the predefined state s_p is consistent with the way how an object is modeled in the design tool, e.g., windows and doors are closed. To really calculate the regions that, for instance, may be covered by a door in a state, the data on the panel extent (*IfcDoorPanelProperties*) and the frame properties (*IfcDoorLiningProperties* and *IfcDoor*) can be applied. In the example scenario of Section 3.6, we only take into account the stable state of an object, as may be modeled within a structural design tool. For instance, the object space does not cover any space occupied by, e.g., deformable or reconfigurable parts of an object.

S2. Operational Space The operational space of an object, henceforth *operational space*, refers to the region of space that an object requires to perform its intrinsic function that characterizes its utility or purpose. For example, for a door that may be opened in one direction, the operational space characterizes the region of space required to facilitate the free movement of the door between, and including, the fully-opened and fully-closed states. For example, if the operational space of a door and a window would overlap, these two may collide if both are open at the same time, resulting in damages. Similarly, the operational space of a rotating surveillance camera is characterized by the angular degrees of movement, which its controllers are capable of. The operational space comprises all space an object may cover regarding all states it can be potentially in:

$$opspace(o) = \bigcup_{s \in \mathcal{S}_o} obspace(o, s)$$

S3. Functional Space The functional space of an object, henceforth simply *functional space*, refers to the region of space surrounding an object within which an agent must be located to manipulate or physically interact with a given object. The functional space is not necessarily similar to the object's convex hull; however, in some cases involving arbitrarily shaped concave objects where interaction is limited to a pre-designated intrinsic front, this space could tend to be more or less equivalent to the object's convex hull. We denote the functional space of an object o by the function $fspace(o)$ – the precise geometric interpretation of this function being determinable in domain specific or externally defined

ways depending on issues such as object granularity, the scale of the ambient environment being modeled, etc. The functional space of an object is dependent on the object o itself, the current state $s \in \mathcal{S}_o$ an object is in, the capabilities of agent a, and the function f, i.e., an action the agent wants to perform on the object. The set of all objects is given by $\mathcal{O}$, the set of all agents by $\mathcal{A}$, and the set of all available functions $\mathcal{F}_a^o$. Thus, we define functional space in $\mathcal{P} = \mathbb{R}^n$ as:

$$\mathit{fspace}(o, a, f, s) = \{p | p \in \mathcal{P} \wedge a \text{ is within range to perform } f \in \mathcal{F}_a^o \text{ on } o \text{ in } s\}$$

As there are arbitrary numbers of agents, functions, and states, it is in many cases impossible to give metrical definitions for combinations of them. Nevertheless, in cases considered here a detailed description of $\mathit{fspace}(.)$ is not necessary to know. Therefore, we use the union $\mathit{fspace}(o)$ of all functional spaces encompassing functions, agents, and states:

$$\mathit{fspace}(o) = \bigcup_{a \in \mathcal{A}} \bigcup_{f \in \mathcal{F}_a^o} \bigcup_{s \in \mathcal{S}_o} \mathit{fspace}(o, a, f, s).$$

S4. Range Space The (sensory) range space, henceforth *range space*, refers to the region of space that lies within the scope of sensory devices such as motion, temperature, heat, humidity and fire sensors, infrared and laser scanners, cameras, and so forth. In order to fulfill functional requirements, it might be necessary to either directly or indirectly ensure certain spatial relationships between a sensor's range space and other artifacts and objects. For instance, it is desirable that the range space of temperature sensor should not overlap with that of a heating device, and desirably, the metric distance between the *object space* of the heating device and the range space of the heating sensor be at a certain minimum for a given room layout. The range space of a sensor in a particular state s is denoted by $\mathit{range}(o, s)$. If the sensor is reconfigurable, the maximal range by the sensor can be given by:

$$\mathit{range}_{max}(o) = \bigcup_{s \in \mathcal{S}} \mathit{range}(o, s).$$

Since this paper is restricted to static environments, as they exist on a work-in-progress design, we do not model different states over time. As such, the range of a sensor in the state is given by: $\mathit{range}(o, s) = \mathit{range}(o)$. Simply, for a stationary sensor o, $\mathit{range}(o) = \mathit{range}(o, s) = \mathit{range}_{max}(o)$.

Explicit characterizations of spatial artifacts are necessary to enforce structural and functional constraints (e.g., Section 1.1, C1–C3) during the AmI design process, especially when the environment is intended to consist of a wide-range of sensory apparatus (cameras, motion sensors, etc.).

3.3 Spatial Artifacts: Concrete Goemetric Interpretations in R^2

As illustrated in Section 3.1, fine-grained semantic distinctions at the level of the spatial ontology (i.e., the conceptual space) and the capability to define their precise geometric interpretation by domain-specific parameters (i.e., the quantity space) are both necessary and useful to stipulate spatial and functional constraints during the design phase. Figure 2 provides a detailed view on the different kinds of spaces we introduced in Section 3.2 for R^2. Although all illustrations in this paper deal with 2D projections of 3D information, note that for the general case, there is no difference between the different characterizations

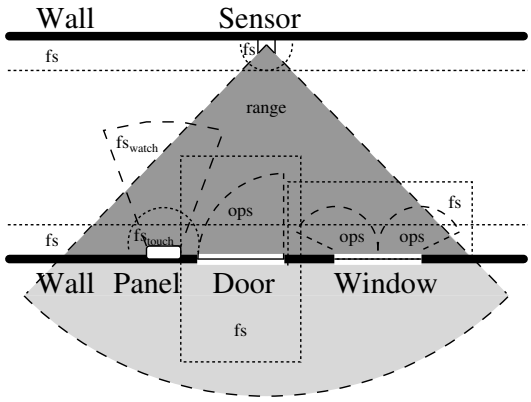
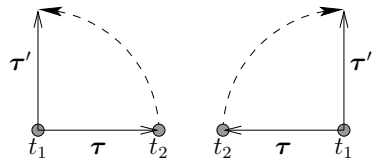


Fig. 2: Differences between functional space (fs, dotted lines), operational space (ops, dashed lines), and range (gray regions surrounded by dashed line).

in functional, operational, and range spaces – they all refer to a physical spatial extension in R^n. However, these do differ with respect to their ontological characterizations within the modular spatial ontologies (Section 3.1), and in the manner of deriving their respective geometric interpretations in R^n. These interpretations differ and depend on, in addition to an object's inherent spatial characteristics (e.g., size and shape), on one or more additional parameters, as elaborated in (G1–G4):

G1. Object Space As walls, panels and sensors are, in general, not reconfigurable or deformable, the object space for the spatial entities is just the space occupied by them, which can be derived on the basis of IFC data. In contrast, doors and windows are reconfigurable. The most natural state of these components is assumed to be *closed* and thus, the object space is defined by the space covered in this state. This also complies with the modality by which doors and windows are depicted in architectural drawings.

G2. Operational Space As doors and windows are reconfigurable they also possess an operational space (dashed line). In Fig. 2 we depict a single panel, right swing door which is defined by its two corner points $t_1 = \binom{x_1}{y_1}$ and $t_2 = \binom{x_2}{y_2}$ with t_1 denoting the position of the hinge. The respective vectors are denoted by $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$. Data necessary for deriving these points are given in the IFC building entities, e.g., *IfcDoorStyle* (type of door) and *IfcDoor* (swing direction). For example, the type of door at hand is referred by *SINGLE_SWING_RIGHT*. We represent the closed door by the vector $\boldsymbol{\tau} = \binom{x_\tau}{y_\tau}$. Currently, opening angles are not represented explicitly in IFC2x3. For reasons of simplicity, we assume a maximum opening angle of 90° for doors. We represent the maximally open door by $\boldsymbol{\tau}'$, which is in our specific case equal to the normal vector $\boldsymbol{\tau}^n$ of $\boldsymbol{\tau}$. The specific direction of $\boldsymbol{\tau}^n$ is defined by the opening direction and hinge position of the door (Fig. 3). Then, the operational space of the door comprises any point between the two vectors $\boldsymbol{\tau}$ and $\boldsymbol{\tau}'$ starting at t_1, which is a sufficient description to calculate any overlap or containment with other regions. In Fig. 2, this results in the quadrant depicted by ops. The two parts of the window may be opened by 135°, which results in the operational space in the manner as depicted.



(a) single panel, left swing door (b) single panel, right swing door

Fig. 3: Operational space calculations for a single panel, left or right swing doors.

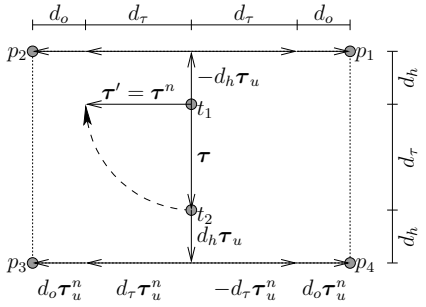


Fig. 4: Functional space calculations for a single panel, right swing door.

G3. Functional Space Regarding different types of agents the area for possible interactions may vary. For humans, e.g., functional space for touching a wall is within the range of an approximate arm length. For artificial agents like robots or semi-autonomous wheelchairs the concrete value may vary. In Fig. 2 we assumed this range by $40cm$ regarding walls. For doors and windows we approximate the space where they can be opened or closed by rectangular shapes. We denote the length of $\boldsymbol{\tau}$ by d_τ and the related unit vector by $\boldsymbol{\tau}_u$. Additionally, we define reachability distances d_h (to the left and right of the door) and d_o (in front and behind the opened door). Based on this information we can derive the four corner points p_i of a rectangle representing $\mathit{fspace}(\mathit{Door})$ by:

$$p_i = \boldsymbol{t}_{\lceil \frac{i}{2} \rceil} + (-1)^{\lceil \frac{i}{2} \rceil} d_h \boldsymbol{\tau}_u + (-1)^{\lfloor \frac{i}{2} \rfloor}((d_\tau + d_o)\boldsymbol{\tau}_u^n) \text{ with } i \in \{1,2,3,4\}.$$

We illustrate this formula in Fig. 4. In Fig. 2, the door panel is one meter wide and we defined $d_h = d_o = 20cm$. For doors or windows where opening angles larger than 90° are possible, the point calculations depend on further aspects. To give an impression we refer to Fig. 5. Note that the shape of the functional spaces may need to be refined depending on the agents and their intended functions with respect to an object at hand (Section 3.2). For example, one must be in close proximity to a panel in order to touch it. In Fig. 2 this is depicted by a semi-circle. In contrast, with respect to an agent who wants to watch the content, this definition is insufficient with respect to the function. The viewing angle (available in technical documentation) and the maximum distance the content can be detected must be considered. This distance may vary with size and the height at which it is fitted to the wall. In Fig. 2, we depicted $\mathit{fspace}(\mathit{panel}, a, \mathit{watch}, s)$ assuming a viewing angle of 20° and a visibility distance of 1.5 meters.

G4. Range Space In general, sensors have angles and maximum measurement distances defined which serve as a direct basis for extracting range spaces. Again, two vectors can be determined to calculate the overlap with other spaces with respect to the sensor's position in the environment. For a camera, for example, these values are the field viewing angle and the camera's resolution compared to the visual angle of objects to detect. In our example, a sensor angle of 110° and a maximum distance of 4 meters are assumed. The light-gray area at the bottom is also part of the range space but will not be accessible by sensors which rely on visibility, e.g., laser or sonar, because it is behind a wall as seen from the sensor. Nevertheless, other sensors, such as magnetic fields or WLAN, overbear these

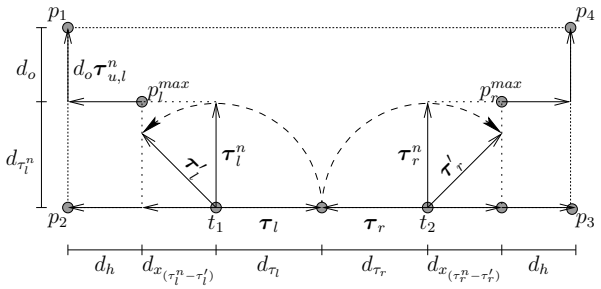


Fig. 5: Operational and functional space calculations for a vertical double panel window with a maximum opening angle of $135°$.

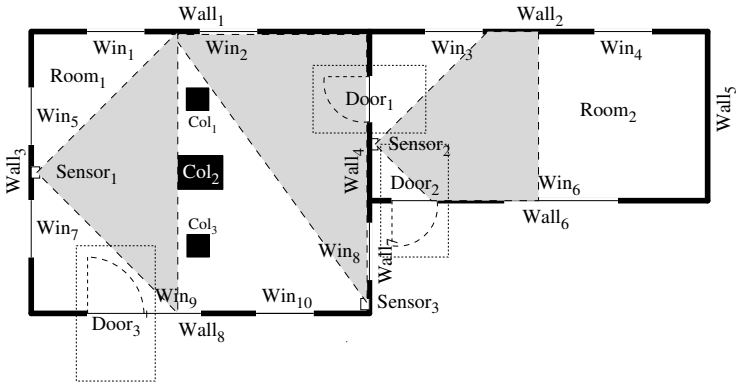


Fig. 6: Exemplary design with Windows, Doors, Columns, and Sensors

barriers and interaction with an agent is possible although positioned behind the wall.

3.4 Spatial Objects and Artifacts: An Example Grounding

In our simple example scenario, depicted in Fig. 6, we only use a small subset of the available components. The scenario consists of eight walls, ten windows, three doors, three columns, and three sensors. For to reasons of clarity only the operational and functional spaces of doors as well as the range spaces of the sensors are presented. Due to the connectivity relation between the walls two rooms are constituted ($Room_1$: Walls 1, 4, 7, 8, and 3; $Room_2$: Walls 2, 5, 6, and 4). Additionally, we have to calculate the spatial artifacts defined in Section 3.2. The different artifacts are calculated automatically in the qualitative module (cf. Section 3.1) from the geometrical data based on predefined formulae as presented in Section 3.3. The necessary data is available in the IFC representation. Additionally, the connectivity structure of walls is given in IFC. If open rooms, i.e., rooms not completely enclosed by walls, are given this can be additionally modeled by zones (*IfcZone*) or spaces (*IfcSpaces*) Thus, we can assume that sufficient metric data is given for any object to calculate RCC-8 relations. For example, the metric representation for Room_1 and Col_2 is given by:

$$
\begin{array}{ll}
 & (\text{ROOM } \text{Room}_1\ (\text{Wall}_1\ \text{Wall}_4\ \text{Wall}_7\ \text{Wall}_8\ \text{Wall}_3)) \\
\Rightarrow & (\text{ROOM } \text{Room}_1\ ((0\ 0)(6\ 0)(6\ 3)(6\ 5)(0\ 5))) \\
 & (\text{COL } \quad \text{Col}_2 \quad ((2.6\ 2.2)(2.6\ 2.8)(3.4\ 2.8)(3.4\ 2.2)))
\end{array}
$$

Following the formulae and definitions in Section 3.3 (with $d_h = d_o = 0.2$ meters), *opspace*() and *fspace*() can be derived easily. For example, the door panel of Door_1 is 0.8 meters long and $t_1 = (6, 0.8)$ ($t_2 = (6, 1.6)$). For *opspace*(Door_1) follows: $\boldsymbol{\tau} = \binom{0}{0.8}$ and $\boldsymbol{\tau}' = \binom{-0.8}{0}$. For *fspace*($\text{Door}_1$) follows: $p_1 = (7, 0.6), p_2 = (5, 0.6), p_3 = (5, 1.8)$ and $p_4 = (7, 1.8)$.

Based on the metrical data, the qualitative model consisting of topological relationship between two regions can be derived. For example, as no point of the region defined by Col_2 (including the boundary) is outside the region or touches the boundary defined by Room_1, Col_2 is a *non-tangential proper part* of Room_1. Additionally, the functional space of Door_1 overlaps with the range spaces of Sensor_1 and Sensor_3. This is reflected in the SBox by:

$$
\begin{array}{llll}
(\ \texttt{rcc-related} & \text{Col}_1 & \text{Room}_1 & :\text{NTPP}\ ) \\
(\ \texttt{rcc-related} & \text{Door}_1\text{\_fs} & \text{Sensor}_1\text{\_rs} & :\text{PO}\ ) \\
(\ \texttt{rcc-related} & \text{Door}_1\text{\_fs} & \text{Sensor}_3\text{\_rs} & :\text{PO}\ )
\end{array}
$$

These calculations are performed for all building entities such that a complete qualitative spatial model with RCC-8 relations is available.

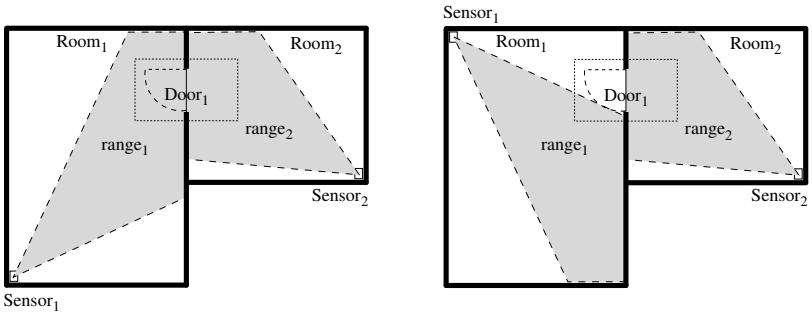
3.5 Requirements Constraints for AmI

Given the Integrated Representation of Section 3.1, particular requirements for AmI environments can be defined. The requirements are formalized by constraints within and across the ontologies. The Integrated Representation itself merely defines general relationships across modules. As such, it supports spatial and terminological reasoning by constraints across modules on a general level. This reflects the modular nature of our ontological representation distinguishing spatial perspectives. An example of such a constraint is the requirement that all M2.Door in the Building Architecture ontology are composed of one M1.Door in the quantitative layer (formulated in Manchester Syntax [16], namespaces are added as prefixes):

Class: construction:Door
SubClassOf: compose exactly 1 arch:Door

Based on such general relationships, the AmI Requirements ontology is formalized. It specifies particular restrictions for building automation in the architectural design process. Hence, it needs to be adjusted to particular building requirements depending on formal structural, functional and potentially other aspects.[2] Particular requirements for our example scenario are based on smart office environments. An example of such a constraint is the requirement that all functional spaces of doors should be a proper part of some range space of some motion sensors. In more detail, it has to be ensured that different motion sensors properly perform a 'handshake' when monitoring persons changing rooms. Within the Building Architecture ontology, the requirement is specified as follows:

[2] Section 5 discusses our outlook on other constraints that may be formalized within the requirements ontology.



(a) Requirement Consistency (b) Requirement Inconsistency

Fig. 7: A two room scenario with the requirement that the door must be supervised by sensors, i.e., the functional space must be completely covered by some sensor range (not necessarily only from a single sensor).

Class: arch:DoorFunctionalSpace
SubClassOf: arch:FunctionalSpace,
rcc:properPartOf some (arch:MotionSensorRangeSpace)

Here, the requirement is defined on the basis of the quality space module (M.2). The Building Architecture ontology defines the classes for functional spaces and range spaces, while the RCC-8 Relations ontology provides the region-based relations. Requirements that take into account different modules, however, are also specified. The requirement that, for instance, all buildings have an intelligent navigation terminal that provides building information for visitors is defined in the following requirement constraint:

Class: arch:Building
SubClassOf: rcc:inverseProperPartOf min 1 (arch:Display
and (ir:conceptualizedBy some physObj:NavigationTerminal))

In this example, the classes Building and Display are defined in the qualitative model and NavigationTerminal is defined in the conceptual model. Their connection conceptualizedBy is defined in the Integrated Representation (ir), while the constraint on the class Building is formalized in the AmI Requirements ontology.

Besides ontology design criteria of such a modular representation, our modeling also shows practically adequate formalizations of spatial entities from the different perspectives. In the quantitative module four walls of a room might actually be modeled by more than four walls. For instance, in the example case in Fig. 6, Wall_4 and Wall_7 of Room_1 are distinguished as two walls in the quantity module. Both walls, however, are mapped to one instance of a Wall in the quality space module. This wall constitute the Room_1 counterpart in the quality space. Note that this mapping is only applicable in this example. In general, it is also possible to define more than four wall instances in the quality space module that may constitute a room. The distinction is then directly supported by the modular structure of the ontologies.

3.6 Requirements Consistency: An Example Scenario

We use spatio-terminological reasoning to ensure that the requirements specified at the AmI design phase are satisfied. For this purpose, the reasoner proves the consistency of the ABox (terminological instances) according to definitions of the TBox and the consistency of the SBox (spatial instances) according to RCC-8 relations. An actual floor plan representation can then be analyzed whether it fulfills the requirements that are defined by the AmI Requirements ontology. Fig. 7 illustrates two alternatives of a selected part of a floor plan (the floor plan illustrations are reduced for simplicity, e.g., windows are omitted).

The examples are specified in the quantity space module on the basis of their IFC data. The classes Room, Sensor, Wall, and Door of the Building Construction ontology are used to instantiate the rooms $\mathsf{M3.Room}_1$ and $\mathsf{M3.Room}_2$, the sensors $\mathsf{M3.Sensor}_1$ and $\mathsf{M3.Sensor}_2$, the door $\mathsf{M3.Door}_1$, and several walls (omitted for simplicity). These instances have to be connected to their counterparts in the quality space module. The instances and their relations are then specified in the Building Architecture (M.2) ontology:

Individual: arch:Room1
Types: arch:Room
Facts: rcc:externallyConnectedTo arch:Room2

Individual: arch:DoorFunctionalSpace1
Types: arch:DoorFunctionalSpace
Facts: rcc:particallyOverlaps arch:SensorRange1
Facts: rcc:particallyOverlaps arch:SensorRange2
⋮

AmI requirements are then be satisfied by proving consistencies of TBox, ABox, and SBox, outlined herein. Note that the consistency itself is proven by using the DL reasoner RacerPro. For completeness, we also describe an example for a TBox consistency proof, albeit that is not directly connected to our specific example scenarios:

TBox Inconsistency: An inconsistency in the TBox is used to determine whether or not the AmI requirements specified by a designer may possibly be fulfilled by a model per se. In the Integrated Representation ontology, the relation isConceptualizedBy, for instance, may define an injective mapping between M2.Room in the qualitative module and M3.RoomType in the conceptual module (the latter defines specific rooms, such as kitchen, office, laboratory, etc.). Assuming that another relation conceptualize in the AmI Requirements ontology allows several mappings between M3.RoomType and M2.Room and that this relation is defined as the inverse relation of isConceptualizedBy, reasoning over the TBox would then detect an inconsistency in the ontology definition itself.

ABox Inconsistency: Inconsistencies in the ABox arise from instances of the ontology that are not compliant with the ontological constraints, both spatial and otherwise. In our case, the instances reflect information of a floor plan while the constraints reflect AmI requirements. An inconsistency in the ABox then implies that the floor plan does not fulfill the AmI requirements. A requirement constraint could be that that all rooms should be equipped with doors

that satisfy certain (metric) traversibility criteria (e.g., with respect to human, wheel-chair, robot movement). The spatial-centric nature of SBox inconsistency (discussed next) notwithstanding, the reasoning pattern involved there is essentially the same as that for the ABox. Hence further details of ABox inconsistency are excluded herein.

SBox Inconsistency: An inconsistency in the SBox identifies those instantiations that are not compliant with RCC-related constraints. In our scenario, it indicates that a floor plan does not satisfy the qualitative spatial requirements. The AmI Requirements ontology, for instance, constrains that individuals, e.g., humans, can be monitored while they leave or enter rooms at any time (cf. C3 in Section 1.1). For this purpose, all doors have to be monitored and therefore all functional spaces of doors have to be a proper part of some sensor range. In Fig. 7(a), the functional space of Door_1 is a proper part of the union of the range spaces of Sensor_1 and Sensor_2. It therefore satisfies the requirements and is proven to be consistent. This can be verified with RacerPro by proving that no instance exists ('NIL'), that is a functional space of a door and not a non-tangential proper part of the range space of a particular motion sensor. The query in RacerPro infers this result:

```
? (retrieve (?*X ?*Y) (and (?X DoorFunctionalSpace)
                           (?Y MotionSensorRangeSpace)
                           not (?*X ?*Y :ntpp)))
> NIL
```

The same request with the example in Fig. 7(b), however, infers that Door_1 is not a non-tangential proper part of some sensor ranges. The example is therefore inconsistent with respect to the AmI requirements. In summary, concrete examples of architectural building plans have to satisfy the requirements given by the AmI Requirements ontology by proving the consistency of their ontological instantiations in the TBox, ABox and SBox.

4 Discussion and Related Work

The field of ambient intelligence has found wide-spread commercial acceptability in the form of applications in the smart environment (e.g., homes, offices) domain [24, 33]. The field has also witnessed considerable inter-disciplinary interactions, hitherto not conceived, from several spheres in artificial intelligence. The design and implementation of this new generation of smart environments demands radically new modeling techniques right from the early design phases. It is necessary for a designer to explicitly communicate the spatial and functional requirement constraints, directly and indirectly related to the perceived smartness of the environment, to the design tool being utilized. Further, it is necessary that such communication accrue in a way that is consistent with the inherent semantic and qualitative manner in which the requirements are conceptualized by the expert. Albeit differing in application and approach, similar sentiments are expressed in [1, 2]. A formal state-based approach for design reasoning that is structures and functions is proposed in [1]. In this approach, structures are defined as states and operations on them are defined as functions. Reasoning is then formulated as an interaction between the two. For the domain of architecture design, this approach has been taken further to create a process by which

requirements can be converted into working design solutions through front-end validation [2]. Although the studies in [1, 2] are different in approach, i.e., we utilize formal methods in ontological and spatial reasoning, the motivations of both from a general architectural design viewpoint remain the same, i.e., to reduce design errors and failures by iterative design validation and verification, and from our AmI design perspective, also to ensure that a work-in-progress design fulfills the functional requirements in order for it to be able to deliver the perceived *smartness*. The crux of such a design approach is that it becomes possible to automatically validate the designer's *conceptual space* against the precisely modeled work-in-progress *quantity space* of the design tool. The operationalization of such a design approach and intelligent assistance capability is the objective of our research, and this paper is a foundational contribution in that regard.

Spatio-terminological reasoning is a well-founded approach for integrated reasoning about spatial and descriptive terminological information [11, 12]. Applications of this paradigm in the GIS domain also exist, e.g., [32] and [18]. The first approach follows the idea of the Semantic Web [5] in the context of GIS applications, i.e., users are able to formulate queries with respect to temporal, spatial, and environmental aspects. The approach of BUSTER [32] aims to improve search options by enriching their data with semantic information. Although we use a combination of spatial and terminological reasoning as well, our focus is on the analysis of consistencies in architectural floor plans anchored in the field of AmI environments. We specify architectural data models in a modular ontological way and reason over concrete instantiations of building representations with spatial objects and artifacts.

Closely related to the architecture data interoperability standard utilized in this work, namely the IFC, is the Building Information Model (BIM) [8]. BIM is an emerging and all-encompassing technological framework that *enables* users to integrate and reuse building information and domain knowledge pertaining to the entire life cycle of a building. The concept of Green Building [14], also connected to the BIM, aims at creating structures and utilizing standards and policies that are environmentally responsible and resource-efficient from a sustainability viewpoint. The concept extends throughout a building's complete life-cycle encompassing the design, construction, operation, maintenance, renovation and deconstruction phases. We further touch upon the importance of such emerging standards and connections with our work in whilst positioning our ongoing work in Section 5.

5 Summary and Outlook

In this paper, we propose, formalize, and demonstrate the application of formal knowledge and spatial modeling constructs, and the paradigm of integrated spatio-terminological reasoning in the domain of smart environment design, or more generally AmI design. The proposed application enables the capability to ensure that semantically specified functional requirement constraints by an AmI designer are satisfied by a metrically modeled work-in-progress design such as a floor plan. The paper presented an example scenario in the context of an architectural data interoperability standard, namely the IFC, and the state-of-the-art design tool ArchiCAD. The formal representation and reasoning components utilized the OWL DL fragment of the ontology modeling language and the

spatio-terminological reasoner RacerPro, and the Region Connection Calculus as a basis of spatial information representation and reasoning.

There are two main areas for further research that our project has adopted. Along the practical front, we are investigating the integration of our approach with a light-weight indoor-environment design tool, namely Yamamoto [29], which offers built-in capabilities for annotating geometric entities at the quantity space with semantic information. From a theoretical viewpoint, we are extending the approach to serve not only a diagnostic function, but also to provide the capability to explicitly prescribe potential ways to resolve inconsistencies within the work-in-progress design. As another line of work, we note that within an *architecture design tool*, metrical changes in the structural layout, which result in qualitative changes along the *conceptual space* of the designer, directly or indirectly entail differing end-product realizations in terms of building construction costs, human-factors (e.g., traversability, safety, productivity, personal communication), aesthetic aspects, energy efficiency, and long-term maintenance expenses thereof based on present and perceived costs. We propose to extend our approach toward the estimation of such material and non-material costs that arise solely by minor conceptual and spatial variations within a design. It is envisioned that these extensions will be achieved within the framework of emerging standards and frameworks such as BIM, IFC and GreenBuilding, and integrated within a state-of-the-art design tool. In a more general direction, we also have to investigate spatial design considerations that derive from cognitive or practical requirements, i.e., whether the design is cognitively adequate. This could, for instance, be realized by simulations that analyze how people interact with the designed environment. Finally, from a rather long-term viewpoint, we regard it interesting to directly collaborate with professional architects via a dialog interface for the communication of design descriptions and constraints with the system. All suggested extensions remain focused to the domain of *smart space* design.

Acknowledgements: We gratefully acknowledge the financial support of the DFG through the Collaborative Research Center SFB/TR 8, projects R3-[Q-Shape] and I1-[OntoSpace]. Additionally, the first author also acknowledges funding by the Alexander von Humboldt Stiftung, Germany. We also thank Bernd Krieg-Brückner and John Bateman for fruitful discussions and impulses. Educational licenses have been utilized for ArchiCAD and RacerPro.

Bibliography

[1] Ö. Akin. Architects' reasoning with structures and functions. *Environment and Planning B: Planning and Design*, 20(3):273–294, 1993.

[2] Ö. Akin and I. Özkaya. Models of design requirement. In *Proceedings of the 6th International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Landgoed Avegoor Ellecom, The Netherlands, 2002.

[3] J. Augusto and D. Shapiro, editors. *Advances in Ambient Intelligence*, volume 164 of *Frontiers in Artificial Intelligence and Applications*. IOS, 2007.

[4] J. C. Augusto and C. D. Nugent, editors. *Designing Smart Homes, The Role of Artificial Intelligence*, volume 4008 of *Lecture Notes in Computer Science*. Springer, 2006.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[6] M. Bhatt and F. Dylla. A qualitative model of dynamic scene analysis and interpretation in ambient intelligence systems. *International Journal of Robotics and Automation: Special Issue on Robotics for Ambient Intelligence*, 2009. (to appear).

[7] M. Day. The Move to BIM with ArchiCAD 12, 23 August, 2008. AEC Magazine.

[8] C. Eastman, P. Teicholz, R. Sacks, and K. Liston. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Frontiers in Artificial Intelligence and Applications. Wiley, 2008.

[9] T. Froese, M. Fischer, F. Grobler, J. Ritzenthaler, K. Yu, S. Sutherland, S. Staub, B. Akinci, R. Akbas, B. Koo, A. Barron, and J. Kunz. Industry foundation classes for project management - a trial implementation. *ITCon*, 4:17–36, 1999. `www.ifcwiki.org/`.

[10] R. Grütter, T. Scharrenbach, and B. Bauer-Messmer. Improving an RCC-derived geospatial approximation by OWL axioms. In A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, editors, *The Semantic Web - 7th International Semantic Web Conference*, pages 293–306. Springer-Verlag, 2008.

[11] V. Haarslev, C. Lutz, and R. Möller. Foundations of spatioterminological reasoning with description logics. In *Proceedings of Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 112–123. Morgan Kaufmann, 1998.

[12] V. Haarslev and R. Möller. Description logic systems with concrete domains: Applications for the semantic web. In F. Bry, C. Lutz, U. Sattler, and M. Schoop, editors, *Proceedings of the 10th International Workshop on Knowledge Representation meets Databases (KRDB 2003), Hamburg, Germany, September 15-16, 2003*, volume 79 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[13] V. Haarslev, R. Möller, and M. Wessel. Querying the semantic web with Racer + nRQL. In *Proceedings of the KI-2004 International Workshop on Applications of Description Logics (ADL'04)*, 2004.

[14] K. Hall, editor. *Green Building Bible: In Depth Technical Information and Data on the Strategies and the Systems Needed to Create Low Energy, Green Buildings*, volume 2. Green Building Press; Fourth edition, 2008.

[15] J. Hois, M. Bhatt, and O. Kutz. Modular ontologies for architectural design. In *4th Workshop on Formal Ontologies Meet Industry*, 2009.

[16] M. Horridge and P. F. Patel-Schneider. Manchester OWL syntax for OWL 1.1, 2008. OWL: Experiences and Directions (OWLED 08 DC), Gaithersberg, Maryland.

[17] I. Horrocks, O. Kutz, and U. Sattler. The Even More Irresistible SROIQ. In *Knowledge Representation and Reasoning (KR)*. AAAI Press, 2006.

[18] K. Kovacs, C. Dolbear, and J. Goodwin. Spatial concepts and OWL issues in a topographic ontology framework. In *Proc. of the GIS*, 2007.

[19] O. Kutz, D. Lücke, and T. Mossakowski. Heterogeneously Structured Ontologies—Integration, Connection, and Refinement. In T. Meyer and M. A. Orgun, editors, *Advances in Ontologies. Proceedings of the Knowledge Representation Ontology Workshop (KROW 2008)*, pages 41–50. ACS, 2008.

[20] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. $\mathcal{E}$-Connections of Abstract Description Systems. *Artificial Intelligence*, 156(1):1–73, 2004.

[21] T. Liebich, Y. Adachi, J. Forester, J. Hyvarinen, K. Karstila, and J. Wix. *Industry Foundation Classes: IFC2x Edition 3 TC1*. International Alliance for Interoperability (Model Support Group), 2006.

[22] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. Ontologies library. WonderWeb Deliverable D18, ISTC-CNR, 2003.

[23] B. Motik, P. F. Patel-Schneider, and B. C. Grau. OWL 2 Web Ontology Language: Direct Semantics. Technical report, W3C, 2008.

[24] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, 2004.

[25] W. Pike and M. Gahegan. Beyond ontologies: Toward situated representations of scientific knowledge. *International Journal of Human-Computer Studies*, 65(7):659–673, 2007.

[26] C. Ramos, J. C. Augusto, and D. Shapiro. Ambient intelligence: The next step for artificial intelligence. *IEEE Intelligent Systems*, 23(2):15–18, 2008.

[27] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning (KR'92)*, pages 165–176, San Mateo, 1992. Morgan Kaufmann, San Mateo.

[28] S. Staab and R. Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.

[29] C. Stahl and J. Haupert. Taking location modelling to new levels: A map modelling toolkit for intelligent environments. In M. Hazas, J. Krumm, and J. Krumm, editors, *Location- and Context-Awareness*, pages 74–85. Springer-Verlag, 2006.

[30] N. A. Streitz, A. Kameas, and I. Mavrommati, editors. *The Disappearing Computer, Interaction Design, System Infrastructures and Applications for Smart Environments*, volume 4500 of *Lecture Notes in Computer Science*. Springer, 2007.

[31] M. Uschold and M. Grüninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–155, 1996.

[32] U. Visser. *Intelligent Information Integration for the Semantic Web*, volume 3159 of *Lecture Notes in Computer Science*. Springer, 2004.

[33] G. M. Youngblood and D. J. Cook. Data mining for hierarchical model creation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37(4):561–572, 2007.