Modular Ontologies for Architectural Design

Joana HOIS, Mehul BHATT, and Oliver KUTZ
SFB/TR 8 Spatial Cognition
University of Bremen, Germany
{joana, bhatt, okutz}@informatik.uni-bremen.de

Abstract. The design of architectural environments has to take into account various sources of heterogeneous information. Not only quantitative spatial constraints and qualitative relations but also functionally-dependent and abstract conceptualizations are relevant aspects for architectural design. We aim at a modular ontological approach based on the theory of $\mathcal{E}$-connections to formally present and bring together these different perspectives on the domain. Modularity here allows a flexible integration of the various sources while keeping their thematically different aspects apart. We show how modular ontologies reflect the domain for architectural design and how they can be applied to smart office environments in ambient intelligence.

Keywords. Ontologies, Modularization, Heterogeneous Information, Architectural Design, Ambient Intelligence

Introduction

Professional architecture design tools are primarily concerned with the ability to develop models of spatial structures at different levels of granularity. They range, for example, 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 building, one may model various spatial elements representing doors, windows, rooms, etc. These elements are based on 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 [3]. Furthermore, and partly as a consequence, these tools also lack the ability to exploit the cognitive expertise that a designer is equipped with. Our approach utilizes ontologies to incorporate semantics at different layers, e.g., the conceptual or mental space of the designer and qualitative abstractions of quantitatively modeled designs.

In this paper, we propose an approach that enables design tools to represent information about architectural environments by combining different perspectives on the domain. In particular, we endeavor to formalize these different and heterogeneous aspects involved in the architectural design process by applying methods from ontological modularity. Here, we address the various aspects described from thematically different perspectives that provide a particular *view* on the domain [28] given by particular ontologies. Ontologies formalize a specific domain and they consist of a taxonomy of classes, relations between these classes, and axiomatizations of classes and relations. They are used as a method for making explicit what is already known implicitly, and they aim at interoperability and re-usability among different sources of information. In the case of architectural design systems, ontologies can provide a formalization of their domain-specific semantics accordingly.

One major topic in the field of *formal ontology* developments concerns ontological modularity and ontology mapping [32]. We pursue this modularity strategy by taking into account the content of ontologies. Here, modular ontologies support design clarity by specifying the different perspectives on a domain. i.e., each modularly-designed ontology provides the semantics for a particular view. As a result, the different ontologies can be connected by integration and can interact in a meaningful way.

In essence, we aim at applying modularity inspired by the theory of $\mathcal{E}$-connections to modular perspectives on architectural design. The ontologies are not only separated into modules logically, but they also reflect different thematic modules. They are applied to industrial standards for architectural design and tools, which are introduced in the next section. Section 2 gives an overview of modularization in ontological engineering and how it is applied to architectural design. We present our modular ontologies in Section 3, followed by an application scenario for architectural design in ambient intelligence in Section 4.

1. Architectural Design Tasks

As the design of specialized spatial environments such as smart buildings and homes starts to become mainstream and economically viable for a larger consumer base, it is expected that architecture design projects involving the design and implementation of such environments will adopt a radically different approach involving the use of formal knowledge representation and reasoning techniques [4, 2]. It is envisioned that a smart environment will be designed from the initial stages to aid and complement the requirements that characterize its anticipated functional or intelligent behavior [1, 3]. A crucial element that is missing in architectural design pertains to formal modeling, i.e., representation and reasoning of architectural structures. Indeed, as all architectural design tasks pertain to a spatial environment, formal representation and reasoning along terminological and spatial dimensions can be a useful way to ensure that the designed model satisfies key requirements that enable and facilitate its intended form and function. Among other things, the ontological modularization proposed in the paper is a key aspect to operationalize this design approach in the initial architectural modeling phase.

We demonstrate our proposed approach in the context of an industrial standard for data representation and interchange in the architectural domain, namely the Industry Foundation Classes (IFC) [10]. This data exchange format is based on object classes, such as *IfcDoor*, *IfcRoof*, or *IfcShapeAspect*. The classes provide information about metrical data, relations, and dependencies between classes. IFC's design techniques are also enabled by architectural design tools, such as ArchiCAD [8], which also supports export capabilities in XML and an IFC compliant binary format. The non-proprietary data model of the Industry Foundation Classes (IFC) [23] aims at fostering interoperability in the building industry by reflecting building information. While 2D or 3D CAD models are based merely on metrical data for geometric primitives, e.g., points and lines, IFC's object classes add semantics to these primitives, i.e., object classes are not only defined by their metrics but also by their inherent relationships to other classes. In our work we use the latest release IFC2x3 TC1 [23], which specifies 653 *building entities* and several *defined types*, *select types*, and *enumerations*. The latter are used for specifying properties and relationships of building entities. Section 3 presents the way IFC's data model is integrated into our modular approach.

2. Ontological Modularity

Modularity has become one of the key issues in ontology engineering. Research into aspects of modularity in ontologies covers a wide spectrum. [32] give a good overview of the breadth of this field. Three orthogonal questions 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? (*modularization 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 differing notions of module and modularity in ontologies and arising reasoning problems. We finally outline the kinds of modules and modular reasoning problems that we encounter in our particular application scenario, the formal specification of architectural designs.

2.1. The Dimensions of Ontological Modularity and Formal Reasoning

The main dimensions of ontological modularity and the respective (automated) reasoning challenges are:[1]

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 [11]. Tool support for such translations is, for instance, provided by the HETS system [26].

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 $\mathcal{OWL}$ 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. A lot can be learned in this respect from techniques developed for (algebraic) specification in software engineering. When applied to ontology engineering, this provides a systematic account that parallels structuring techniques from algebraic specification with typical problems found in ontology design [21].

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 [19, 21]. 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 [24].

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.

[1] This is, of course, not an exhaustive analysis.

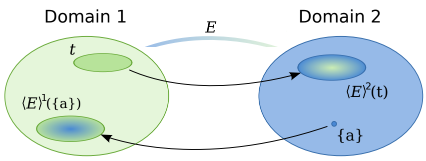

Figure 1. A two-dimensional connection.

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. Deciding conservativity is undecidable for first-order logic and many expressive description logics (DL) [32]. However, there are general algorithmic solutions for less expressive DLs [19], and also solutions for more expressive DLs based on syntactic criteria [6]. The simplest case of a conservative extension is a *definitonal 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 [9] and aligning [34] 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 $\mathcal{O}$ is any operation by which O_1, O_2 are 're-interpreted' from the (global) point of view of $\mathcal{O}$. This has been utilized in the approach of [31] (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 $\mathcal{O}$, where the models of $\mathcal{O}$ 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.

In $\mathcal{E}$-connections, 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, an 'abstract' object o of a description logic DL_1 can be related via a relation R to its life-span in a temporal logic T (e.g., a set of time points) as well as to its spatial extension in a spatial logic S (e.g., a set of points in a topological space). 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.[2]

$\mathcal{E}$-connections have also been adopted as a framework for the integration of ontologies in the Semantic Web [12], and, just as DLs themselves, offer an appealing compromise between expressive power and computational complexity: although powerful

[2]Thus analyzed, the main difference between distributed description logics (DDLs) [5] and various $\mathcal{E}$-connections then lies in the expressivity of the 'link language' $\mathcal{L}$ connecting the different ontologies.

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 [22], $\mathcal{E}$-connections in general add substantial expressivity and interaction to the components. 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. Fig. 1 displays the connection of two ontologies, with a single link relation E.

2.2. Modularity in Architectural Design Specification

The aspects of ontological modularity that we employ in order to realize the envisioned application to architectural design are manual alignments, conservative (definitional) extensions, $\mathcal{E}$-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.

Definitional Extensions: New concepts are added to the DOLCE-Lite ontology in the *Physical Object* ontology (see Section 3.2) by a definitional signature extension. Moreover, we add new concepts to the spatial relations in the *Building Architecture* ontology, 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 $\mathcal{E}$-connecting the aligned vocabulary along newly introduced link relations and appropriate linking axioms. As dedicated reasoners for $\mathcal{E}$-connections are not available at the moment, we realize this scenario by using $\mathcal{E}$-connections in a way that allows a complete encoding of the semantics of $\mathcal{E}$-connections into $\mathcal{OWL}$ DL (compare also [12, 7]): (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 $\mathcal{E}$-connection operators we use DL existential and universal restrictions on these link relations.

Global extensions of integrated representations: New constraints are added on top of the integrated representation by $\mathcal{E}$-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 4).

3. Modular Ontologies for Architectural Design

In the following, we present different ontologies developed for architectural design.[3] First, we outline general ontological formalization aspects underlying their development. We subsequently introduce the modularly specified ontologies and what they specify, in particular from metrical, quantitative, and different conceptual perspectives, while taking

[3] see http://www.ontospace.uni-bremen.de/ontology/modSpace/AmI.html

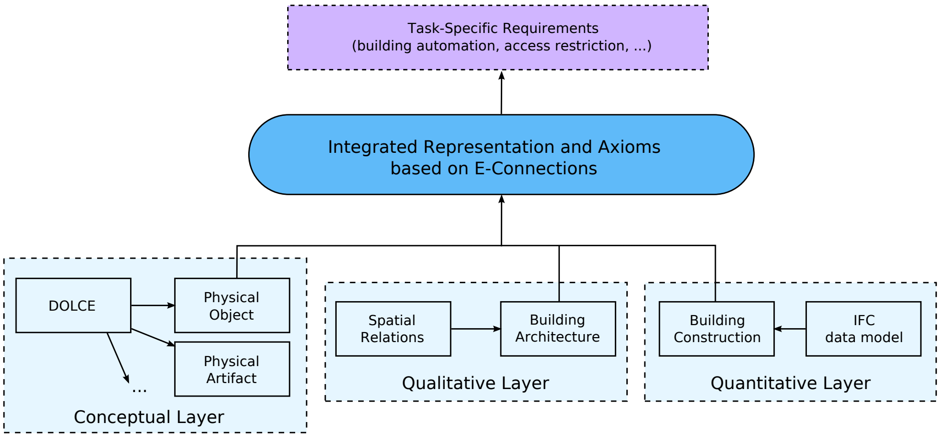

Figure 2. Different layers contributing heterogeneous aspects of architectural design.

into account formal ontological design criteria and modularity issues. Finally, results with respect to expressivity and reasoning aspects are evaluated.

3.1. General Specification Aspects

Although ontologies can be defined in any logic, we focus here on ontologies as theories formulated in DL, supported by the web ontology language $\mathcal{OWL}$ DL 2 [27]. DL distinguishes between TBox and ABox. The TBox comprises all class and relation definitions, while the ABox comprises all instantiations of these classes and relations. Even though ontologies may be formulated in more or less expressive logics, DL ontologies have the following benefits: they are widely used and a common standard for ontology specifications, they provide constructions that are general enough for specifying complex ontologies, and they provide a balance between expressive power and computational complexity in terms of reasoning practicability [18].

Reasoning over the TBox, which defines the categorization and axiomatization of the domain, 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, which defines instances and their relations in a specific model of the TBox, allows, for instance, to classify instances or to determine additional relations among instances. In case of architectural design, the domain of buildings and their characteristics and constraints have to be defined. The requirements for classes and instances of concrete buildings can then be axiomatized.

In addition to reasoning over ontologies (TBox) and their instances (ABox), however, spatial reasoning is of particular interest, as spatial positions of entities and their spatial relations between each other are highly important to describe the environment. Here, we apply a specific feature of the reasoning engine RacerPro [16] that supports region-based spatial reasoning directly by the so-called SBox [14], which provides reasoning based on the region connection calculus [30], in particular RCC-5 and RCC-8 (see 4.2).

3.2. Thematically Different Ontologies for Architectural Design

In the architectural design process different criteria can be closely grouped into modules that reflect different topics. An overview is illustrated in Figure 2. Basic information with respect to architectural design is metrical and geometrical data, i.e., a *quantitative layer*. It particularly reflects metrical data of construction elements of a floor plan for specific buildings, such as Wall, Door, or Window. From this constructional perspective, entities are specified on the basis of their metrical aspects, such as size, position, or opening angle. In contrast, more abstract spatial relations between such entities are indicated by a *qualitative layer*, which specifies dependencies and spatial constraints. Even though it defines similar entities to those from quantitative information, their characteristics are based on spatial relationships to other entities. While, for instance, quantitative aspects of a Wall merely determine the wall's height and length, qualitative aspects of a Wall describe its bounding of rooms or its connection to ceilings and floors. Finally, a *conceptual layer* in architectural design specifies architectural entities and phenomena as such. All entities are given by their intrinsic features, e.g., a Wall is described by its material, color, or style. All layers can be specified by ontologies. Some of them partially extend or re-use existing ontologies. They are then connected with each other by formalizing link relations across ontologies, which results in the *Integrated Representation*. This ontology can be further extended for task-specific aspects to provide a certain purpose.

3.2.1. Quantitative Layer

In our use case, we describe information on construction plans on the basis of industrial building components specified by the IFC data model (cf. Section 1). Sample classes from IFC are mirrored by this first layer of architectural information. Constructional elements of buildings are specified by BuildingElement in the *Building Construction* ontology of the quantitative layer. The properties, they have to specify, are at least their height, length, and width. The formulation (in Manchester Syntax [17]) is:

```
Class:        BuildingElement
SubClassOf:   height exactly 1 Float, length exactly 1 Float, width exactly 1 Float
```

Information provided by this quantitative layer is related to concrete floor plans and instantiated accordingly. It can also constrain minimal or maximal sizes of certain entities (e.g., rooms or corridors) on a metrical basis. Another dimension is part of to this representational layer could also be added, namely data properties specified with concrete domains [15]. For instance, the values of the size of certain entities, such as windows and doors can be bound to certain upper and lower limits, in order to ensure accessibility.

3.2.2. Qualitative Layer

Information on spatial-functional architectural entities and their qualitative spatial relationships are formalized in the *qualitative layer*. Region-based relationships are defined for entities in this module. The entities are similar to those specified in the *Building Construction* ontology, but they specify non-metrically determined entities, e.g., functional aspects of entities. A Door, for instance, is specified as a connection between rooms and corridors, it is spatially connected to adjacent entities, and it is also provided by a spatial region that indicates its spatial-functional access.

The *Building Architecture* ontology of this module offers region-based spatial relationships, as indicated by the region connection calculus RCC-8 [30], by re-using the RCC ontology that is introduced in [13]. The eight region-based relations are defined in a property hierarchy. These properties are applied to the entities specified in the qualitative layer. For example, the kinds of adjacent entities of a Door are constrained as follows:

```
Class:        Door
SubClassOf:   BuildingElement,
              rcc:externallyConnectedTo some (Wall or Window), ...
```

As outlined in Section 3.1, RCC-8 relations and their implications are provided by the SBox of RacerPro. It allows spatial reasoning over these relations. Region-based dependencies on certain architectural entities can be analyzed and architectural requirements can be proven accordingly (cf. [4]).

3.2.3. Conceptual Layer

In the *conceptual layer* architecture-related entities are based on their idiosyncratic characteristics, i.e., they are specified by their properties and axioms without any contextual or embedded aspects. For example, particular subclasses of Door are specified here, such as SwingDoor, RevolvingDoor, or SlidingDoor. Entities from the conceptual layer can be related to floor plans but also even more abstract entities, e.g., functions, costs, or actions. Such conceptual ontologies can extend foundational ontologies, which provide an abstract foundation for specifying specific domain entities and relations. One example of such a formal ontology is DOLCE [25].

For the architectural design case, we have extended DOLCE-Lite [4] formulated in $\mathcal{OWL}$ DL in order to provide a categorization of architectural entities. The resulting *Physical Object* ontology particularly refines *physical endurants* [25]. Specific qualities can also be described, e.g., the material type of a Window is here defined as one of aluminum, steel, wood, or plastic:

Class:	Window
SubClassOf:	BuildingConstruct, DOLCE-Lite:has-quality exactly 1 Material

We could also consider more abstract categories than constructional aspects, e.g., *physical artifacts* as introduced in [33]. If such *intentionalities* can be defined for an entity, they could be formalized accordingly. However, the specification of this kind of information is left for future work.

3.2.4. Integrated Representation

The previous three modules are now combined and linked in the *Integrated Representation* ontology. As described in Section 2, the theory of $\mathcal{E}$-Connections is encoded by providing an additional layer of axioms. Even though all three ontologies are imported into the *Integrated Representation*, only link relations between classes from different ontologies are defined here. As a first step, the classes from the different ontologies are made disjoint, as the different perspective given by the ontologies are not supposed to be ‘matched’. Similar classes from different ontologies are not mapped to each other, instead a link relation between them has to be defined.

DisjointClasses:	DOLCE-Lite:particular, buildingArchitecture:Functional_Structure, buildingConstruction:Architectural_Feature

Note, that these three classes are ‘artificially constructed’ top node classes from the respective ontologies (namespaces indicate their different origins). As a result, the classes from different ontologies are defined as disjoint sets, i.e., no instance can be specified in more than one of them. In addition, the *Integrated Representation* ontology does not define any new classes or subclasses. Instead, it specifies correspondences between classes from different ontologies by defining link relations, which are specified as bijective mappings using cardinality constraints. For example, a specific class in the quan-

[4]http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl

titative layer, such as Door in the Building Construction ontology that reflects metrical information about doors, has its correspondence in the qualitative layer, such as Door in the Building Architecture ontology that reflects connections to spatially adjacent entities. The link relation compose specified in the *Integrated Representation* ontology defines the relation from the quantitative layer to the qualitative layer, and its inverse relation isComposedOf vice versa:

```
ObjectProperty:   compose
Domain:           buildingConstruction:Architectural_Feature
Range:            buildingArchitecture:Functional_Structure
InverseOf:        isComposedOf
Class:            buildingConstruction:Door

SubClassOf:       compose exactly 1 buildingArchitecture:Door
Class:            buildingArchitecture:Door

SubClassOf:       isComposedOf exactly 1 buildingConstruction:Door
```

A similar link relation is defined for entities from the qualitative to the conceptual layer. This relation is again bijective and specified by conceptualizedBy and conceptualize accordingly. Given this integrated layer that defines the link relations between the different ontologies, extensions of this layer can axiomatize specific requirements to model certain design criteria.

3.2.5. Task-Specific Requirements

Based on the integrated representation layer, this layer adds additional definitions and constraints to architectural information available by the different layers described above. It formulates requirements that have to be satisfied by a concrete floor plan. These requirements describe certain tasks or purposes a specific design has to meet. They can, for instance, determine regulations for user access, automatic control of electronic devices, or monitoring and surveillance tasks. As an example, one ontology of this task-specific layer aims at specifying requirements for *ambient intelligence* (see Section 4). It therefore defines constraints on assistance systems as well as motion sensors and visual sensors that are supposed to monitor movements inside buildings.

These constraints are formalized in the *task-specific* ontology as ontological constraints on classes from the different modules re-using link relations of the integrated representation. For instance, the requirement that all buildings have an intelligent navigation terminal that provides building information for visitors is defined by the following requirement constraint:

```
Class:         buildingArchitecture:Building
SubClassOf:    rcc:inverseProperPartOf min 1 (buildingArchitecture:Display
                   and (integratedRepresentation:isConceptualizedBy some
                   physicalObject:NavigationTerminal))
```

4. Application Scenario: Ambient Intelligence

The field of ambient intelligence (AmI) has become a rapidly evolving area in research, business, and administration [29]. Its aim is to enrich the environment of daily life aspects by providing automatic control and decision support with respect to environmental conditions. AmI is intended to support humans in achieving their everyday activities. Areas of applications in ambient intelligence particularly focus on work and home scenar-

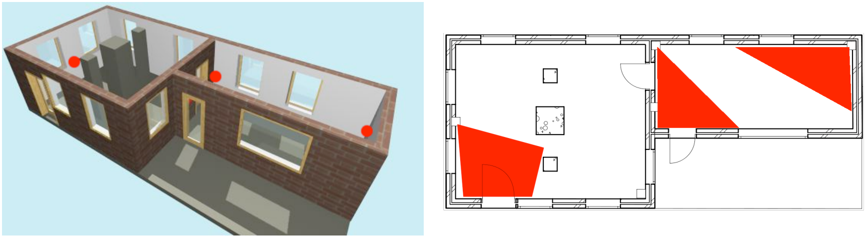

Figure 3. Example of an architectural design, in which motion sensors (illustrated as red dots in the left image) provide monitoring support (sensor ranges are illustrated as red areas in the right image).

ios, they address issues of transportation and navigation, accessibility, tourism, elderly and health care [29]. The architectural design process of AmI environments therefore has to take into account relevant information on these criteria.

Formalizing design criteria and architectural information is one major issue in the field of ambient intelligence [20]. Given our modular ontological representation, these different aspects are formalized and related with each other. The representation supports the design process by analyzing whether design criteria are satisfied, e.g., particular AmI requirements.

4.1. Use Case: AmI Requirements

The modular ontologies for architectural design can be used in the design process of AmI environments for defining requirements. In Figure 3, an example of an architectural design is illustrated. The underlying IFC data of this example is instantiated as an ABox in the *Building Construction* ontology. ABox Reasoning over their metrical information analyzes whether the instances satisfy the ontological constraints.

The integrated representation requires the instances in *Building Construction* to be linked to instances in the *Building Architecture* ontology. The ABox of this latter ontology has to be instantiated accordingly. Region-based spatial relations between these instances are calculated on the basis of metrical information and the qualitative layer then defines the topological relationships. Examples of such calculations are outlined in [4]. The qualitative relations are specified by the RCC-8 relations as pre-defined in the SBox of RacerPro, and SBox consistency is evaluated.

Moreover, the integrated representation requires the instances of the qualitative layer to be linked to instances in the conceptual layer. Particular PhysicalObjects are instantiated and related to the qualitative layer. Global consistency can be proven by ontological reasoning in the integrated representation. In addition, particular requirements for AmI environments in the *task-specific requirements* ontology can be analyzed. In our example, 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. Whether such requirements are satisfied by a model is proven.

4.2. Preliminary Results: Expressivity and Reasoning

We have developed and applied the integration of modular ontologies for architectural design to ensure consistency requirements in AmI environments. Several axioms that have to be satisfied by concrete floor plans can be adjusted and refined in the *task-specific* ontologies. Link relations can be used to constrain relations between instances from different ontologies, and ABox consistencies can be easily analyzed by ontological reasoning. In detail, we use ontological alignments, conservative and global extensions, and $\mathcal{E}$-connections for the ontological representation.

Furthermore, we use reasoning provided by RacerPro for SBox reasoning over RCC-8 relations. Hence, floor plans can be analyzed with respect to their consistency. They are directly given by ontological consistency proofs. In cases that only affect RCC-8 relations and instances in the *Building Architecture* ontology, a consistency proof exclusively in this module may be sufficient. If constraints in the *task-specific requirements* ontology use link relations of the *integrated representation*, only global consistency can be proven, although the integrated representation is based on modularly developed ontologies.

Furthermore, as the ontologies are formulated in $\mathcal{OWL}$ DL, their expressiveness is limited to axioms formulated in description logics. Even though this is sufficient for our purpose and representation of architectural design, actual $\mathcal{E}$-connections in the integrated representation have to be encoded because the classes from different ontologies are defined as disjoint sets. Also, the *task-specific requirements* layer has to import all other ontological modules, which increases complexity and which is unnecessary in most cases.

5. Conclusions and Future Work

In this paper, we have presented modularly developed and $\mathcal{E}$-connected ontologies for architectural design. Ontological reasoning is used for proving consistency of task-specific requirements. Applying formal methods for modularity supports the ontological design and the representation of different perspectives on the domain. The application scenario outlines how a metrically modeled work-in-progress design is enriched with ontologically specified requirement constraints.

The integration of different architectural information, however, may be extended in several directions: concrete domains can be used in the *quantitative* layer in order to provide detailed axiomatizations of certain entities, e.g., the minimum and maximum size of steps of a staircase. Several ontologies can be defined for *task-specific* requirements, e.g., energy saving, navigation, home entertainment, or emergency situations. The *qualitative* layer can be extended by further spatial relations than region-based relations, e.g., orientation, distance, or shape-based relations. These extensions depend on specific reasoning support and are left for future work.

Acknowledgements: We gratefully acknowledge the financial support of the DFG through the COLLABORATIVE RESEARCH CENTER SFB/TR 8, projects I1-[ONTOSPACE] and R3-[Q-SHAPE]. Additionally, the second author also acknowledges funding by the ALEXANDER VON HUMBOLDT STIFTUNG, GERMANY.

Independent online content accompanying paper in particular and the overall project are available at: `http://www.cosy.informatik.uni-bremen.de/staff/bhatt/cosit09-www/`

References

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

[2] 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.

[3] 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).

[4] M. Bhatt, F. Dylla, and J. Hois. Spatio-terminological inference for the design of ambient environments, 2009. (submitted for publication).

[5] A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *Data Semantics*, pages 153–184, 2003.

[6] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular Reuse of Ontologies: Theory and Practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.

[7] B. Cuenca Grau and O. Kutz. Modular Ontology Languages Revisited. In *Proc. of IJCAI'07 Workshop SWeCKa*, 2007.

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

[9] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.

[10] 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/`.

[11] J. A. Goguen and R. M. Burstall. Introducing institutions. In E. Clarke and D. Kozen, editors, *Proc. Logics of Programming Workshop*, pages 221–256. Springer, 1984.

[12] B. C. Grau, B. Parsia, and E. Sirin. Combining OWL ontologies using $\mathcal{E}$-Connections. *Journal Of Web Semantics*, 4(1):40–59, 2006.

[13] 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.

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

[15] 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, *Proc. of the 10th International Workshop on Knowledge Representation meets Databases (KRDB 2003)*, volume 79 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

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

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

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

[19] R. Kontchakov, F. Wolter, and M. Zakharyaschev. Can You Tell the Difference Between DL-Lite Ontologies? In *Proc. of KR'08*, pages 285–295. AAAI, 2008.

[20] A. Kung. Architecture and design patterns for ambient intelligence: an industry perspective. In A. Maña and C. Rudolph, editors, *Developing Ambient Intelligence, Proc. of the International Conference on Ambient Intelligence Developments*, pages 55–67. Springer, 2008.

[21] 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. Proc. of the Knowledge Representation Ontology Workshop (KROW 2008)*, pages 41–50. ACS, 2008.

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

[23] 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.

[24] C. Lutz. Description Logics with Concrete Domains—A Survey. In *Advances in Modal Logics Volume 4*. King's College Publications, 2003.

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

[26] T. Mossakowski, C. Maeder, and K. Lüttich. The Heterogeneous Tool Set. In O. Grumberg and M. Huth, editors, *TACAS 2007*, pages 519–522. Springer-Verlag Heidelberg, 2007.

[27] B. Motik, P. F. Patel-Schneider, and B. C. Grau. OWL 2 Web Ontology Language: Direct Semantics. Technical report, W3C, 2008. `http://www.w3.org/TR/owl2-semantics/`.

[28] W. Pike and M. Gahegan. Beyond ontologies: Toward situated representations of scientific knowledge. *International Journal of Man-Machine Studies*, 65(7):674–688, 2007.

[29] C. Ramos. Ambient intelligence - a state of the art from artificial intelligence perspective. In J. Neves, M. F. Santos, and J. Machado, editors, *Progress in Artificial Intelligence, 13th Portuguese Conference on Aritficial Intelligence, EPIA 2007 Workshop Proc.*, pages 285–295. Springer, 2007.

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

[31] M. Schorlemmer and Y. Kalfoglou. Institutionalising Ontology-Based Semantic Integration. *Journal of Applied Ontology*, 3(3), 2008.

[32] H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors. *Modular Ontologies*. Springer, 2009. To appear.

[33] L. Vieu, S. Borgo, and C. Masolo. Artefacts and roles: Modelling strategies in a multiplicative ontology. In C. Eschenbach and M. Grüninger, editors, *Formal Ontology in Information Systems (FOIS'08)*, pages 121–134. IOS Press, 2008.

[34] A. Zimmermann, M. Krötzsch, J. Euzenat, and P. Hitzler. Formalizing Ontology Alignment and its Operations with Category Theory. In *Formal Ontology in Information Systems (FOIS'06)*, pages 277–288, 2006.