Spatial Computing for Design

— An Artificial Intelligence Perspective

Mehul Bhatt and Christian Freksa
SFB/TR 8 Spatial Cognition
University of Bremen
Germany
www: `http://www.sfbtr8.spatial-cognition.de/`
e-mail: {bhatt, freksa}@sfbtr8.uni-bremen.de

ABSTRACT

The articulation of the *Science of Design* by Herbert Simon and the paradigmatic relevance of Artificial Intelligence in that context are closely intertwined topics: Simon elaborates the 'Sciences of the Artificial' in the context of the design of artefacts. Situated in this AI-centric view of design, we characterize "*spatial computing for design*" as a specialisation concerned with the development of the general representational and computational apparatus necessary for solving modelling and reasoning problems in spatial design. Several representation and reasoning problems are discussed in the backdrop of relevant examples involving the formal modelling of *structural form* with respect to a desired / anticipated *artefactual function*. The discussion, although applicable to any spatial design activity, is grounded in the domain of assistive decision-support in the context of a conventional computer-aided architecture design workflow.

Keywords: knowledge representation and reasoning; spatial and conceptual reasoning; diagnosis and recommendation; decision support for design architecture / spatial design; knowledge engineering for design; design semantics; computer-aided architecture design.

1 Artificial Intelligence and Design

The significance and the paradigmatic relevance of Artificial Intelligence in Modern Design is intertwined with Herbert Simon's original articulation of the *Science of Design* [Simon 1969; 1996] and in the words of Baldwin [2007], Simon's interpretation of design as a "*decision-making process under constraints of physics, logic and cognition*". This view of the scientific design process underlies much of what artificial intelligence has to offer by way of its formal representational and computational apparatus to the domain of design.[1] From a topical viewpoint, the knowledge representation and reasoning area within artificial intelligence has been the cornerstone of most formal AI inroads in so far as *problem-solving*

[1] Henceforth, by design we refer to spatial design in general, and in specific to architectural design, which we regard to be an instance of spatial design. By conventional design systems, we refer to computer-aided architectural design (CAAD) tools.

for design is concerned. In the last two decades, several interdisciplinary initiatives comprising of computer scientists, engineers, psychologists and, designers have addressed the application of artificial intelligence techniques for solving problems that accrue at several stages of the design process: design conceptualization, functionality specification, geometric modelling, structural consistency & code-checking, optimization, collaborative (design) workflow management, design creativity, and a plethora of other issues.[2]

Situated within this AI-centric view of the science of design, we present our perspective on *spatial computing for design*. Strongly influenced by the need to formally define, model, and reason about (structural) form & (artefactual) function, our interpretation of spatial computing encompasses three aspects we regard as crucial:

- semantic modelling, spatial abstraction, & multi-perspective representation
- design analysis by inference patterns supporting diagnostic & hypothetical reasoning
- assistive feedback / communication with designers

The aspects deemed essential correspond to problems that accrue within a conventional 'iterative refinement by automated design assistance' workflow, and are identifiable with respect to the *modelling–evaluation–re-design* phases in intelligent design assistance, for instance, as interpreted within a Function-Behaviour-Structure (FBS) [Gero 1990, Gero et al. 1991] model of the design process. With respect to the refinement work-flow, the basic research questions within the context of spatial computing include:

1. Semantics: formal modelling of design requirements, and the role of knowledge engineering in that regard
2. Spatial abstraction: abstraction of CAD-based geometric information into the qualitative domain via the use of formal spatial representation and reasoning techniques
3. Qualitative spatial reasoning: the application of spatial consistency as a basis for checking for design requirement consistency
4. Hypothetical reasoning: the role of hypothetical reasoning (e.g., by abduction) as a means to support a diagnostic and recommendation function within a logic context
5. Assistive feedback: visualisation modalities as a means to interact & communicate assistive feedback with the designer

[2]The journal "Artificial Intelligence for Engineering Design, Analysis and Manufacturing" completed two decades of publishing in 2007 and its anniversary publication is a good overview of the area [Brown 2007, Gero 2007]. A sketch of '*40 years of design research*' is available in [Bayazit 2004]. The collected works of [Akin 1993, Brown 1993, Chandrasekaran 1990, Gero 1990, Gero et al. 1991, Hirtz et al. 2002, Krishnamurti 2006] are a rich source of reference and contextualisation.

The above problem aspects have fuzzy boundaries and many interrelationships. However, the paper attempts to characterize each one of them rather independently via running examples. The paper is organized as follows:

Section 2 is an exposition of the philosophy that underlies our approach to spatial computing. The points raised condition the basic premises of our overall approach, especially our propositions on hypothetical reasoning for design. Section 3 provides an overview of the iterative refinement cycle in design. Here, we exemplify the key aspects of spatial computing for design vis-à-vis the iterative refinement cycle. In Section 4, we define spatial computing and present the issues that we deem to be within its scope. Key representational and computational modalities are discussed, and we also attempt to ground the discussion in Sections 2 and 3 with examples that further illustrate the agenda of spatial computing, and the problems that may be solved therein. This section can be considered to be our statement of the work-in-progress in spatial computing for design. Finally, we summarize in Section 5, and at the same time, also reflect upon some of the issues raised by Gero [1991] in his statement of "*Ten Problems for AI in Design*".

2 The Philosophy of Spatial Computing, for *Spatial Design*

In architectural design, we are faced with structures in physical space. Much of the design considerations in architectural design are directly constrained by intrinsic properties of physical space. Unlike some abstract spaces, the dimensions of physical space are strongly interrelated. This has to do with the fact that the three spatial dimensions are of the same quality: an object that is long in the x-dimension will be long in the y- or z-dimension simply by changing its orientation; one does not have to change the (nature of the) object itself. In color space, for example, we cannot maintain such constancy by moving an object, as each dimension of color space refers to a different quality or feature. On the other hand, the number of spatial dimensions is limited to three; thus, in whatever ways we move objects in space, we will stay within the interrelation of three spatial dimensions.

Besides these intrinsic spatial constraints, we have physical constraints due to mechanical properties of physical objects. In particular, there is a correlation between length, width, and height due to mechanical requirements: longer objects need to be made thicker for maintaining stability properties; thicker objects may become larger to maintain proportions, etc.

Human perception also treats the three spatial dimensions in similar ways: for manipulable objects the perception of length, height, and width can be transformed by changing the orientation of the objects; for large-scale objects like buildings and mountains, the vertical dimension may not be perceived identically to the two horizontal dimensions.

The main message of these considerations is that physical dimensions are strongly interrelated and that physical space is severely constrained. This can be viewed as a strong limitation in comparison to abstract spaces in which arbitrary configurations of feature values and arbitrary transitions between them are conceivable. From the perspective of design, however, these constraints can be considered a great advantage, as they considerably reduce the space of design decisions.[3]

These considerations not only have implications on the spatial structures to be designed but also on the structures of design computers. Todays general purpose computers represent spatial entities and environments in the conceptual framework of unconstrained abstract spaces; thus, the intrinsic properties of physical space must be explicitly coded into the system to make sure physically realizable designs result from the computational process. In other words, computation needs to be invested to reduce the set of conceivable designs to the set of realizable designs. This is not the case when the designer works directly with spatial models, as these maintain the spatial constraints inescapably.

We use the notion of spatial computing in a way that exploits the intrinsic constraints of spatial structures in such that only those structures will be generated that are realizable in physical space and that do not require a computational reduction from conceivable structures to physically realizable structures.

3 Assisted Iterative Refinement in Spatial Design

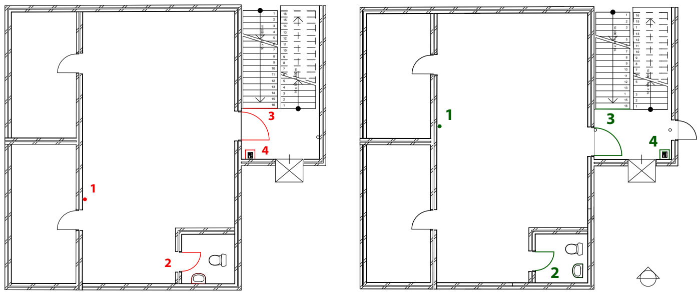
Spatial design as a problem-solving activity typically consists of the Conception – Modelling – Evaluation – Re-modelling cycle. Essentially, a designer in this case is considered to be an *agent of change*, who in the absence of any computational assistance, may be intuitively regarded as traversing a complex configuration space of possibilities, and selecting one course of action (guided by domain knowledge, expertise, cognitive capabilities, specialized requirements, aesthetic preferences and so forth) that produces a desired product / design.

A Design Task. As a basic use-case, consider an architect / engineer specialising in the design and development of building automation systems and smart environments. A typical design challenge would be:

> *Design the layout of an office environment to satisfy structural and functional requirements that collectively aid and complement (and never hinder) the building's automation systems (monitoring devices, sensors, etc.), and which, by implication, facilitate the intended smartness of such automation systems.*

From the viewpoint of the overall design requirements, aspects of this problem explicitly pertain to the functional aspects (e.g., security, privacy, building-automation, accessibility) of the space being modelled, structural code-checking with respect to building regulations, and also possibly specialized client demands. Some example requirements follow in (R1–R3):

[3] Hypothetical reasoning about designs focussing on *what could be* rather than *what is* benefits from rich ontological characterizations along these lines. This is further elaborated on in Section 4.4). Also, see the treatment of aspectualization for architectural design in [Bertel et al. 2004; 2007].



(a) Inconsistent plan (b) Consistent plan

Figure 1.: Design Requirements: Example Spatial Interpretations

R1. certain areas within a building / floor / room should (not) be trackable by sensing devices such as cameras, motion-sensors

R2. regional statutory requirements that stipulate structural constraints and other categorical specifications, e.g., disability access codes, design guides

R3. client specification: as much as possible, the operation of doors should be non-interfering with the functionality of nearby utilities / artefacts

Figure 1 is a schematization of the consistent and inconsistent models of the example requirements / scenarios in (R1–R3). The following aspects, marked as [1–4] in Figures 1(a)–1(b), make the plans of Fig. 1 (in)consistent with respect to (R1–R3):

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

From the viewpoint of spatial computing, one may imagine the search space to consist of spatial configurations— topological, orientational arrangements —and the spatial transformations that are possible, e.g., with respect to a movement taxonomy, as the available actions that produce a re-arrangement. The objective of iterative refinement in general, be it automated or human, is to create consistent models that fulfill the requirements as they are conceived at design time. Albeit a bit limiting, for this particular case, the automation necessary to realize the re-configuration may be identified as a limited form of assistive spatial

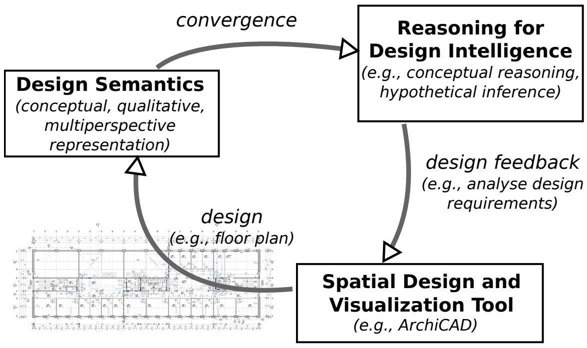


Figure 2.: Iterative Refinement by Intelligent Design Assistance

design intelligence that guides the designer toward a solution that meets the pre-specified requirements, such as those stipulated in (R1–R3).

Automated Design Refinement. Figure 2 illustrates our interpretation of this process of iterative refinement, as it is applicable to the 'spatial computing for design' framework (Section 4) laid out in this paper. The following aspects of iterative refinement (A1–A3) are deemed crucial:

A1. *Modelling – Design Abstraction*: this aspect encompasses issues ranging from semantic specifications, taxonomic representations, qualitative abstractions of geometric models, and modularity of information representation

A2. *Convergence – Reasoning*: this aspect constitutes the various *modes of inference* that constitute the computational manifestations of the assistive design support

A3. *Assistive Feedback – Visualization*: this aspect constitutes mechanisms and modalities to provide diagnostic feedback and other forms of support within a conventional CAAD workflow

Indeed, the possibilities to broaden the interpretation of this manner of intelligent assistance are rather extensive, ranging within a wide array of techniques from the computing, cognitive, psychological, and aesthetic disciplines. Our preliminary focus in spatial computing is centred on spatial cognition, and is guided by the aim to formally and computationally understand the relationship between the "structural form" and "artefactual function" within the domain of spatial design. Further elaborations are presented in Section 4.

4 A Characterization of Spatial Computing for Design

We characterize *Spatial computing for design* in two ways: firstly, by the scientific questions that it must address from a representational and computational viewpoint and their relationships to the domain of artificial intelligence & design in general, and secondly, by the outcomes that a paradigm such as this is expected to produce. Spatial computing for design is defined as:

- that body of work that is concerned with the use of formal methods in knowledge representation and reasoning in general, and terminological and spatial representation and reasoning in specific, for solving problems in modelling (e.g., spatial semantics, modularity, requirement constraints) and validation (e.g., diagnosis, hypothetical reasoning) in the domain of spatial design

- that body of work whose aim is to develop the generic apparatus— application framework, methodology, tool-sets —that may be used as a basis of providing assistive design support within a conventional CAAD-based spatial design workflow

We now elaborate on the representational and computational aspects of the above definition.

4.1 Modelling Form and Function

"*Form follows Function*" [Sullivan 1896] and "*Ornament is Crime*" [Loos 1930]—these two doctrines have been the cornerstones of the modernist tradition in engineering design.[4] Restricting the application of this doctrine to the domain of architectural design, the interpretation that it leads to is that the *structural form*, i.e., *shape*, *layout*, *connectivity*, of a building should be primarily (or more rigidly: *solely*) determined by its practical *function* or *purpose*. Much of the literature in the philosophy of design and architecture [Vermaas et al.], and the ensuing debates thereof, have focused on the semantics of *functions* with respect to design artefacts and the causal link between *form* and *function*, stressing the question of whether or not form should, or indeed does, wholly or in part follows function[5].

Structural Form and Artefactual Function in Spatial Computing: Spatial computing is primarily concerned with the issues surrounding the formal interpretation of the terms "spatial / structural form" and "artefactual function", in particular with respect to the interpretation of these concepts in the context of a CAAD-based workflow. This is crucial, since it is necessary to explicitly put these notions into practice by investigating what precisely does it mean to model *form* and *function* within an intelligent architectural design assistance system. We note some examples:

► **Example 1**. Bremen (Germany) Building code [BremLBO 2003]):

(a). *Staircase / Treppen* (§35 (10), pg. 24):

> "*Steps of a staircase may not be connected directly to a door that opens in the direction of the steps. There has to be a landing between the staircase steps and the door. The length of this landing has to have at least the size of the door width*".

[4] Whereas Louis Sullivan articulated the relationship between of 'Form and Function', the original attribution goes to the 18th century Italian architectural theorist Carlo Lodoli.

[5] Dorst and Vermaas [2005] present a critical review of the Function-Behaviour-Structure model. The discussion sheds useful insights about the nature of form-function

► **Example 2**. US Courts Design Guide 2007 [US GSA 2007]

(b). *Barrier-Free Accessibility* (pg. 4-3):

> *"Courtroom areas used by the public must be accessible to people with disabilities. Private work areas, including the judge's bench and the courtroom deputy, law clerk, bailiff, and court reporter stations, must be adaptable to accessibility. While all judges benches and courtroom personnel stations do not need to be immediately accessible, disabled judges and court personnel must be accommodated"*

(c). *Psychology, Culture and Aesthetics* (pg. 3-1, 4-4):

> *"The architecture of federal courthouses must promote respect for the tradition and purpose of the American judicial process. To this end, a courthouse facility must express solemnity, integrity, rigor, and fairness."*

> *"All architectural elements must be proportional and arranged hierarchically to signify orderliness. The materials employed must be consistently applied, be natural and regional in origin, be durable, and invoke a sense of permanence."*

> *"The height and location of the judges bench expresses the role of the judge and facilitates control of the court. Generally, the judges bench should be elevated three or four steps (21-24 inches or 525-600 mm) above the courtroom wall."*

(d). *Visibility* (pg. 3-2, 16-9):

> *"The entrance or entrance vestibule should be clearly visible and recognizable as such from the exterior of the building. The vestibule should be a minimum of 7 feet in depth and able to handle the flow of traffic at peak times."*

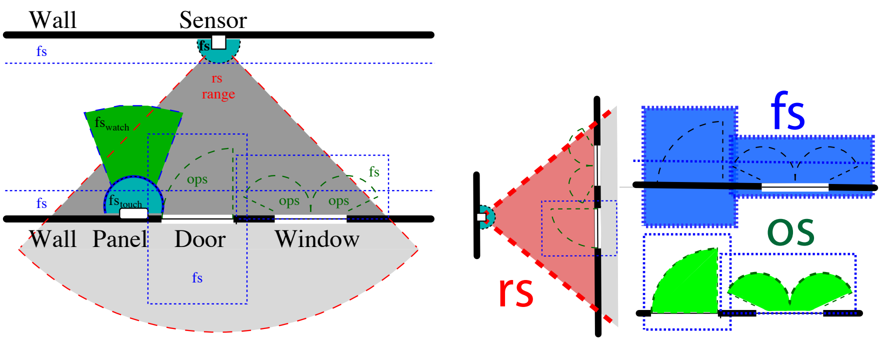
> *"A duress alarm must be easily accessible and visible to all occupants."*

► **Example 3**. A Pattern Language [Alexander et al. 1977]

(e). *Sunny Counter* (pg. 916–918):

> *"Place the main part of the kitchen counter on the south and southeast side of the kitchen, with big windows around it, so that sun can flood in and fill the kitchen with yellow light both morning and afternoon"*

At this stage, we leave the readers with their imagination as to the formal interpretation of the above examples – some have a clear and well-defined spatial structure within a design, whereas others are only indirectly specifiable. Spatial computing in design should be concerned with the extent to which functional aspects such as those exemplified herein could be formally interpreted in strictly semantic and spatial terms; from a computational viewpoint, it is clear that adequate conceptual, spatio-linguistic and qualitative modelling techniques are necessary for representing and reasoning about *design artefacts* and *patterns* entailed by designer expertise.



(a) Implicit Artefacts within a Design

(b) Functional space (fs), OperatioRange space (os), Range space (rs)

***Figure 3.*:** Spatial artefacts are entities, which unlike regular spatial objects, do not have a physical manifestation in reality (or within a design), but need to be treated as such for all practical / reasoning purposes. **Illustration adapted from**: Bhatt et al. [2009]

4.2 Design Artefacts: Conceptualization and Formal Representation

Spatial computing involves an interplay between the designer's conceptualization, the handicaps of the computational constructs of the design tool, and the limitations of the bridges that connect that conceptual with the computational: professional design tools simply 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-centred conceptualization, i.e., *semantically* and *qualitatively*. Modelling for spatial computing in design has to be focussed on representation of *design semantics*, *artefactual modelling* capability and support for *multi-perspective modularity*.

Design Semantics. An expert's design conceptualization is semantic and qualitative in nature—it involves abstract categories such as Rooms, Doors, Sensors and the spatial (topological, directional, etc.) relationships among them, e.g., '*Room A and Room B have a Door in Between, which is monitored by Camera C*'. Whereas this example is rather specific, typical real-world constraints are mostly underspecified or fuzzy (e.g., see Section 4.1). Therefore, any vision of specialised spatial computing for design has to handle to the modelling of designer / design semantics in an explicit manner, e.g., using formal knowledge engineering constructs such as ontology modelling languages.

Spatial Artefacts. A crucial aspect that is missing in contemporary design tools is the support to explicitly characterize the artefactual aspects, and the functional requirements ensuing therefrom, within a design. Semantic descriptions of designs and their requirements acquires real significance when the spatial and functional constraints are among strictly spatial entities as well as abstract *spatial artefacts*. For instance, although it is possible to model the spatial layout of an environment at a fine-grained level, it is not possible to model *spatial artefacts* such as the *range space* of a sensory device (e.g., camera, motion sensor, view-point of an agent), which is not strictly a spatial entity in the form of having a material existence, but needs to be treated as such nevertheless. In general, architectural working designs only contain physical entities. Therefore, it becomes impossible for a designer to model constraints involving spatial artefacts at the design level. For instance, consider the following constraint: '*the motion-sensor should be placed such that the door connecting room A and room B is always within the sensor's range space*'. Bhatt et al. [2009] identify three types of spatial artefacts:

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, or any other entity capable of visual perception. Range space may be further classified into other categories, such as *observational space* (e.g., to model the concept of the *isovist*[6])

Fig. 3 provides a detailed view on the different kinds of spaces we introduced. From a geometrical viewpoint, all artefacts refer to a conceptualised and derived physical spatial extension in $\mathbb{R}^n$. However, they do differ from an ontological perspective and the manner in which their geometric interpretations in $\mathbb{R}^n$ are derived. The derivation of an interpretation may depend on object's inherent spatial characteristics (e.g., size and shape), as well as additional parameters referring to mobility, transparency, etc.

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

As an example, consider a binary relation 'connects' that links entities from the conceptual, qualitative, and quantitative levels of Fig. 4; a Floor at the conceptual level is abstracted

[6] An isovist is the set of all points visible from a given vantage point in space and with respect to an environment [Benedikt 1979].

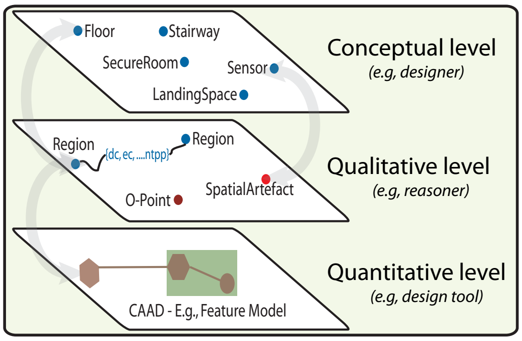


Figure 4.: Multi-Perspective Representation & Modularity

as a Region at the qualitative level of a reasoner and as a ClosedPolygon thereby preserving the geometry at the quantitative level of a CAAD-based feature model:[7].

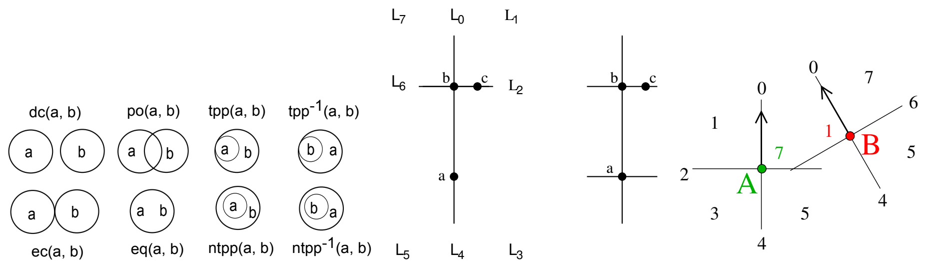
1.	**BinaryLink:**	integration_module:connects
	Domain:	quantitative_level:Architectural_Feature
	Range:	qualitative_level:Functional_Structure
	InverseOf:	integration_module:connectedBy
2.	**Class:**	quantitative_level:ConvexPolygon
	SubClassOf:	connects exactly 1 qualitative_level:Region
3.	**Class:**	conceptual_level:Floor
	SubClassOf:	connects exactly 1 qualitative_level:Region

4.3 Spatial Representation and Reasoning

The field of Qualitative Spatial Reasoning (QSR) investigates abstraction mechanisms and the technical computational apparatus for representing and reasoning about space within a formal, non-metrical framework [Cohn and Renz 2007, Freksa 1991]. Relational formalizations of space and tools for efficiently reasoning with them are now well-established [Renz and Nebel 2007]. In QSR, spatial information representation corresponds to the use of formal spatial calculi such as the Region Connection Calculus [Randell et al. 1992] (RCC), Single-Cross and Double-Cross Calculi (SCC, DCC) [Freksa 1992], Oriented Point Relation Algebra (OPRA) [Moratz 2006] (see Fig. 5).

Within spatial computing for design, the use of formal qualitative spatial calculi and conceptual design requirements serve as a link between the *structural form* of a design and the differing *functional capabilities* that it affords or leads to. Therefore, a very important

[7]The examples are illustrated using a scheme that is close to the so-called Manchester Syntax Horridge and Patel-Schneider [2008] for the description of ontological knowledge in the Web Ontology Language (OWL). The syntax 'M:C' represents a concept 'C' within particular ontological module 'M'. Formal descriptions for these examples may be found in [Hois et al. 2009]



(a) RCC-8 Primitives (b) SCC & DCC Partitioning (c) The $\mathcal{OPRA}_2$ relation $\vec{A}\ {}_2\angle_7^1\ \vec{B}$

Figure 5.: Topological and Orientation Calculi

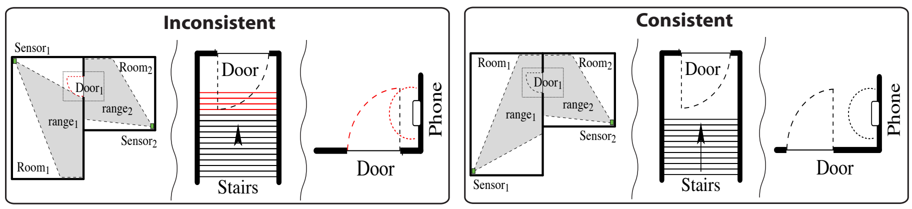


Figure 6.: Design Requirements: Example Spatial Interpretations

goal in spatial computing is to formally and computationally investigate the link between structural forms, as denoted by specific spatial configurations of domain entities, and the behaviours / functions that they are inherently capable of producing with respect to a pre-specified set of requirements conceptually expressed by an architect or a designer.

Artefactual Constraints, Structural Form and Design Function. Spatial artefacts such as those introduced in (A1–A3) are usable towards formulating functional requirement constraints for a work-in-progress spatial design. Constraints, emanating from the requirements such as in in (R1–R3; Section 3) may need to be satisfied by a design:

C1. The FunctionalSpace of the Door of every Office should overlap with the RangeSpace of one or more Camera or MotionSensor.

C2. The StairWay should be topologically non-overlapping with the FunctionalSpace and OperationalSpace of other entities

C3. People should not be harmed by Doors opening up. In general, the OperationSpace of a Door should be non-interfering, i.e., not overlap with the function / operation (i.e., functional/operational space) of surrounding objects.

The schematization in Fig. 6 is a continuation of the example requirements introduced in (R1–R3), and semantically expressed constraints in (C1–C3). To consider two of the three consistent/inconsistent cases from Fig. 6, namely (C1, C3), below is a semantically grounded semi-formal representation of a requirement constraint:

C1.	**Class**:	qualitative_level:DoorFunctionalSpace
	SubClassOf:	qualitative_level:FunctionalSpace, space:topology:properPartOf some (qualitative_level:SensorRangeSpace)

C3.	**Class**:	qualitative_level:PhoneFunctionalSpace
	SubClassOf:	qualitative_level:FunctionalSpace, not (space:topology:overlaps some (qualitative_level:DoorOperationalSpace))

The remaining example from Fig. 6, corresponding to (C2), too may be modelled in a similar manner, namely, as a topological constraint among the primitive conceptual / qualitative / quantitative entities within the design model. Clearly, there are many more possibilities to model requirement constraints on the basis of other aspects of space, e.g., orientation, cardinal directions, metric / fuzzy distances. In this manner of modelling, it must be emphasised that the resulting functional consistency is interpreted strictly with respect to the structural form of the design.

4.4 Design Intelligence - Modes of Inference

The term design intelligence is rather open and subject to diverse interpretations; its scope and definition are only limited by the range of the inference patterns that may be operationalised computationally. From the viewpoint of this paper, we have rather specific inclinations with respect to the reasoning capabilities that must be the focus of spatial computing for design.

Conceptual Reasoning. Conceptual reasoning corresponds to the ontological reasoning patterns that are available within the framework of a terminological reasoning system grounded to the semantics of a *Description Logic* (DL) [Baader et al. 2003]. Ontology reasoning systems such as RACER [Haarslev et al. 2004], PELLET [Sirin et al. 2007] support typical DL inference tasks at the *terminological* (subsumption, satisfiability, equivalence, disjointness) and *instance* levels (instance checking, data consistency, realisation, retrieval). For example:

1. *Retrieval task*: identify all concrete entities / geometric features (e.g., '*polygons*') from instance data coming from a CAAD model that correspond to a design abstraction / artefact such as 'FunctionalSpace' or 'MovableEntity'

2. *Instance checking*: given a set of geometric features within a CAAD model, what is the most general / specific abstract ontological category that the identified feature belongs to from the conceptual / artefactual viewpoint of the designer

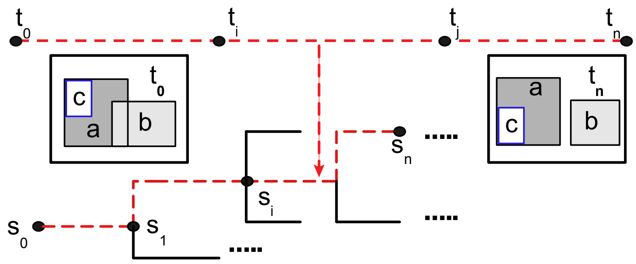


Figure 7.: Branching / Hypothetical Situation Space

From a conceptual reasoning viewpoint, another important reasoning task is determining whether or not the requirement constraints, functional or otherwise, specified by a designer may possibly be satisfied by a model per se. This form of reasoning is useful to check if a given set of design requirement are mutually consistent.

Functional Consistency. The example scenarios in Section 4.3 illustrated the extent and manner in which functional requirement consistency may be modelled with respect to the structural form of design. This is the form of consistency that has been discussed and illustrated throughout this paper. However, the notion of functional consistency transcends beyond the purely spatial aspects of a design, and also includes semi-spatial aspects that include the material and constitution of design artefacts, aspects such as weight, colour, physical characteristics, and artistic aspects that may be beyond the domain of space. Regardless if what precisely what these aspects are, the inference patterns required to ensure functional consistency, in so far as it is formalisable, is essentially some form constraint reasoning approach over a spatial or non-spatial domain, which is the forte of the state-of-the-art in AI research (see Section 5).

Hypothetical Reasoning. Reasoning about conceptual & functional consistency is only a starting point: for spatial computing, the real challenge of intelligent design assistance is the capability to reason about not what is, but instead about what could be. This form of inference is referred to as *hypothetical reasoning*. In general, within a decision-support or design assistance tool, metrical changes in the structural layout or changes in the relative spatial relationships of the design elements – i.e., qualitative changes along the conceptual space of the designer – will directly or indirectly entail differing end-product realizations in terms of spatial design requirements, building construction costs, human-factors (e.g., traversability, way-finding complexity), aesthetics aspects, and energy efficiency and long-term maintenance expenses thereof. As such, commonsensical and hypothetical reasoning at the qualitative level about *physically realizable*[8] and functionally consistent structural forms represents a useful solution approach that is useful for providing the designer with creative design recommendations.

[8] Also related is the commonsensical notion of a *physically realizable situation* defined in terms of physical, compositional and existential consistency of spatial situations [Bhatt 2009; 2010b].

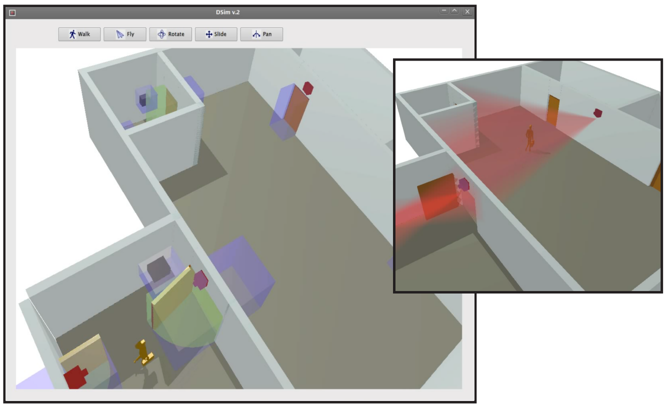


Figure 8.: 3D realizations of the *functional*, *operational* and the *range* spaces of the architectural entities. **System DSim** [Bhatt et al. 2010]

Alternate recommendations are derivable by hypothesizing the possible / potential spatial re-configurations / transformations (e.g., by translation and deformation actions) at the qualitative level; by not discretizing the space and considering the full range of quantitative possibilities, the problem of hypothetical reasoning is in full generality infinitesimal and intractable. As an example, consider the illustration in Fig. 7. The situation-based history $< s_0, s_1, \ldots, s_n >$ represents one path, corresponding to an actual time-line $< t_0, t_1, \ldots, t_n >$, within the overall branching-tree structured situation space that could be representative of a space of design evolutions at the qualitative level. Therefore, the objective of hypothetical reasoning about the '*design space*' is to infer / hypothesize (e.g., by abduction) physically plausible qualitative variations in a design that are also essential or functional requirement fulfilling. Indeed, hypothetical reasoning may also take into consideration domain-specific heuristics / physical attributes that determine aspects such as movability, deformability, stability. Such a logic-based approach may also work as a complementary technique to other approaches such as generative and emergent computations.

4.5 Assistive Feedback Mechanisms – Design Simulation

Assistive feedback mechanisms by visualisation and simulation have to be provided in order to communicate diagnostics and other forms of design support within a conventional CAAD workflow. Conventional CAAD tools have remained focussed on providing capabilities for aesthetically appealing 3D visualisation of floor-plans. State-of-the-art tools also allow easy placement / visualisation of third-part 3D models of common interior artefacts, thereby enhancing the 3D visualisation experience. The human-computer interaction aspects involved in the communication and interaction between the a designer and next-generation CAAD tools is an open topic of research. It is not our objective here to speculate on the future directions of this field of research. The visualisation and simulation aspects pointed out in the following are some benchmarks that have been set for our working prototype DSim [Bhatt et al. 2010]. DSim attempts to operationalize the concept of being able to "*live your design*":

- *Semantic browsing* vis-à-vis the structural hierarchy of the design
- Real-time spatial *artefact simulation* (e.g., sensors, camera; see Fig. 8)
- *Inconsistency pinpointing* at the structural and semantic level
- Hierarchical and *selective zooming* for specific requirements
- Automatic *reconfiguration and placement* of design artefacts

We consider the above features to be crucial and necessary for next-generation CAAD tools that not only support the 2D / 3D spatial modelling, but also provide the conceptual spatial modelling and functional reasoning capabilities, such as those described in this paper.

5 Discussion and Summary

We have addressed two themes in this paper: spatial computing for design on the one hand, and the design of spatial computing itself on the other. The main focus here has been on introducing spatial computing for design as a paradigm, the representational and computational aspects that it needs to addresses as a body of work, and finally the concrete application scenarios that it needs to solve. Our notion of spatial computing (for design) is firmly grounded in the AI / KRR-centred perspective, as enshrined in the initial foundations laid out by early pioneers in the field. Gero [1991] positioned "*Ten Problems for AI in Design*".[9] With respect to the scope spatial computing, as addressed in the present paper, we relate to some of them:

- Representation in design, Design semantics – "*What is it that the designer knows and how do we get a computer to know it*?"
- Inference in design – "*much of design inferencing has to do not only with deductive inference but with abductive inference which is concerned with what might be rather than what is*"
- Combinatorial explosion in design – "*as soon as a system deals with what could be it could go on indefinitely*"

[9] In view of the developments in AI in the last two decades, it is interesting to relate these problems as they existed back then, and as they stand now. We leave this exercise to another paper.

The problem of representation in design and design semantics is related to the modelling of multiple-perspectives and the explicit representation of requirements as per their conceptualization by a designer. The problems of reasoning about what could be and combinatorial explosion in design are two sides of the same coin: hypothetical reasoning (by abduction or otherwise), as positioned in this paper, within a qualitative context, and under additional constraints of physical realizability and architecture domain-specific heuristics is an interesting approach that merits further treatment.

Much has changed in AI since the early 90s. Frame-based systems and semantic networks have evolved into a range of description logic based ontology languages that are tailored to different levels of expressivity and computational properties [Baader et al. 2003]. Practical ontology reasoning systems such as *Racer* [Haarslev et al. 2004] and *Pellat* [Sirin et al. 2007] have also come to the fore. The field of qualitative spatial representation and reasoning has emerged has a new discipline within KRR in the last decade – specialized (infinite domain) spatial reasoning systems *SparQ* [Wallgrün et al. 2007]and *GQR* [Westphal et al. 2009] now support constraint reasoning and additional application-support services that make it possible to model and reason about spatial knowledge in ways that has not been possible before. Similarly, the evolution of Logic Programming (LP) to Constraint Logic Programming (CLP) [Jaffar and Maher 1994] and other powerful computational embodiments of the default and non-monotonic reasoning paradigms by way of Answer-Set Programming (ASP) [Vos 2009] are developments that have only found limited attention in the design community. High-level formalisms to reason about action and change such as the Situation Calculus [McCarthy and Hayes 1969] and the Event Calculus [Kowalski and Sergot 1986] and Fluent Calculus [Thielscher 1998], and other more specialized formalisms also similarly grounded in mathematical logic [Davis and Morgenstern 2004], have progressed to the point where prototypical languages (e.g., Indigolog [Giacomo and Levesque 1999], Discrete Event Calculus Reasoner [Mueller 2006], FLUX [Thielscher 2005]) allow high-level specification and projective / abductive inference capabilities about dynamic process-like phenomena. The new developments open up interesting new possibilities and programming paradigms not only for solving design problems hitherto considered to be computationally intractable, but also for integration, in fundamental ways, of generalised logic-based reasoning on the one hand, and specialized spatial reasoning techniques on the other [Bhatt 2010a].

The progress made in the last two decades within the knowledge representation and reasoning community in general, and the field of spatial reasoning in specific, warrants a re-visitation into the ‘*design as problem-solving*’ approach of Simon [1969]. In spite of garnering initial momentum and interest in the ‘AI for Design’ community, this approach failed to make an impact by way of practical industrial applications. This paper is partly a statement of our work-in-progress, and partly an attempt to revive some of the basic questions underlying AI in / for design in the context of the specialization we refer to as Spatial Computing.

Acknowledgements: We gratefully acknowledge the funding and support by the Alexander von Humboldt Foundation (Germany), and the German Research Foundation (DFG). The paper has immensely benefitted from our discussions and ongoing collaborations with John Bateman, Frank Dylla, Gregory Flanagan, Joana Hois, and Oliver Kutz.

References

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

C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, 1977. ISBN 0195019199.

F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*, 2003. Cambridge University Press. ISBN 0-521-78176-0.

C. Baldwin. Steps toward a science of design. In *NSF Principal Investigators Conference on the Science of Design*, 2007. URL `http://www.people.hbs.edu/cbaldwin/DR2/BaldwinScienceofDesignSteps.pdf`.

N. Bayazit. Investigating design: A review of forty years of design research. *Design Issues*, 20(1), 2004.

M. L. Benedikt. To take hold of space: isovists and isovist fields. *Environment and Planning B: Planning and Design*, 6(1):47–65, January 1979. URL `http://ideas.repec.org/a/pio/envirb/v6y1979i1p47-65.html`.

S. Bertel, C. Freksa, and G. Vrachliotis. Aspectualize and conquer. In J. Gero, B. Tversky, and T. Knight, editors, *Visual and Spatial Reasoning in Design III*, pages 255–279. Key Centre of Design Computing and Cognition, University of Sydney, 2004.

S. Bertel, C. Freksa, and G. Vrachliotis. Aspect-oriented building design: Towards computer-aided approaches to solving spatial constraint problems in architecture. In G. Allen, editor, *Applied spatial cognition: From research to cognitive technology*. Lawrence Erlbaum, Mahwah, NJ, 2007.

M. Bhatt. Commonsense inference in dynamic spatial systems: Phenomenal and reasoning requirements. In I. Bratko and J. Žabkar, editors, *23rd International Workshop on Qualitative Reasoning (QR 09), Ljubljana, Slovenia, June 2009*, pages 1–6, 2009. (Part 1 of 2).

M. Bhatt. Reasoning about space, actions and change: A paradigm for applications of spatial reasoning. In *Qualitative Spatial Representation and Reasoning: Trends and Future Directions*. IGI Global, USA, 2010a. URL `http://www.cosy.informatik.uni-bremen.de/staff/bhatt/seer/Bhatt-2010-RSAC-Book.pdf`.

M. Bhatt. Commonsense inference in dynamic spatial systems: Epistemological requirements. In *FLAIRS Conference: Special Track on Spatial and Temporal Reasoning*. AAAI Press, 2010b. (Part 2 of 2).

M. Bhatt, F. Dylla, and J. Hois. Spatio-terminological inference for the design of ambient environments. In K. S. Hornsby, C. Claramunt, M. Denis, and G. Ligozat, editors, *Conference on Spatial Information Theory (COSIT'09)*, pages 371–391. Springer-Verlag, 2009.

M. Bhatt, A. Ichim, and G. Flanagan. Dsim: A tool for assisted spatial design. In *Proceedings of the 4th International Conference on Design Computing and Cognition (DCC'10)*, 2010. (to appear).

BremLBO. Bremische landesbauordnung, 2003. URL `http://www.bauordnungen.de/html/bremen.html`.

D. C. Brown. Intelligent computer-aided design (1998 revised version). In J.G.Williams and K.Sochats, editors, *Encyclopedia of Computer Science and Technology*, 1993.

D. C. Brown. AI EDAM at 20. *Artif. Intell. Eng. Des. Anal. Manuf.*, 21(1):1–2, 2007. ISSN 0890-0604. doi: http://dx.doi.org/10.1017/S0890060407070011.

D. C. Brown, M. B. Waldron, and H. Yoshikawa, editors. *Intelligent Computer Aided Design, Proceedings of the IFIP WG 5.2 Working Conference on Intelligent Computer Aided Design (IntCAD91), Columbus, OH, USA, 30 September - 3 October 1991*, volume B-4 of *IFIP Transactions*, 1992. North-Holland. ISBN 0-444-81560-0.

B. Chandrasekaran. Design problem solving: a task analysis. *AI Mag.*, 11(4):59–71, 1990. ISSN 0738-4602.

A. G. Cohn and J. Renz. Qualitative spatial reasoning. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007.

E. Davis and L. Morgenstern. Introduction: progress in formal commonsense reasoning. *Artif. Intell.*, 153(1-2):1–12, 2004. ISSN 0004-3702.

K. Dorst and P. Vermaas. John gero's function-behaviour-structure model of designing: a critical analysis. *Research in Engineering Design*, 16(1):17–26, November 2005. doi: 10.1007/s00163-005-0058-z. URL `http://dx.doi.org/10.1007/s00163-005-0058-z`.

C. Freksa. Qualitative spatial reasoning. In D. Mark and A. Frank, editors, *Cognitive and linguistic aspects of geographic space*, pages 361–372. Kluwer, Dordrecht, 1991.

C. Freksa. Using orientation information for qualitative spatial reasoning. In *Proceedings of the Intl. Conf. GIS, From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pages 162–178, Berlin, 1992. Springer-Verlag. ISBN 3-540-55966-3.

J. Gero. Ten problems for AI in design. In *Workshop on AI in Design, IJCAI-91*, 1991.

J. S. Gero. Design prototypes: A knowledge representation schema for design. *AI Magazine*, 11(4): 26–36, 1990.

J. S. Gero. AI EDAM at 20: Artificial intelligence in designing. *AI EDAM*, 21(1):17–18, 2007.

J. S. Gero, K. W. Tham, and H. S. Lee. Behavior: A link between function and structure in design. In Brown et al. [1992], pages 193–225. ISBN 0-444-81560-0.

G. D. Giacomo and H. J. Levesque. An incremental interpreter for high-level programs with sensing. In H. J. Levesque and F. Pirri, editors, *Logical Foundation for cognitive agents: contributions in honor of Ray Reiter*, pages 86–102. Springer, Berlin, 1999.

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.

J. Hirtz, R. Stone, D. McAdams, S. Szykman, and K. Wood. A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*, 13(2):65–82, 2002.

J. Hois, M. Bhatt, and O. Kutz. Modular Ontologies for Architectural Design. In *Proc. of the 4th Workshop on Formal Ontologies Meet Industry, FOMI-09, Vicenza, Italy*, volume 198 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

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

J. Jaffar and M. J. Maher. Constraint logic programming: A survey. *J. Log. Program.*, 19/20:503–581, 1994.

R. Kowalski and M. Sergot. A logic-based calculus of events. *New Gen. Comput.*, 4(1):67–95, 1986. ISSN 0288-3635.

R. Krishnamurti. Explicit design space? *Artif. Intell. Eng. Des. Anal. Manuf.*, 20(2):95–103, 2006. ISSN 0890-0604. doi: http://dx.doi.org/10.1017/S0890060406060082.

A. Loos. Ornament and Crime. *Innsbruck, Reprint Vienna*, 1930.

J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

R. Moratz. Representing relative direction as a binary relation of oriented points. In *ECAI*, pages 407–411, 2006.

E. T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006. ISBN 0123693888.

D. A. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In *KR'92. Principles of Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, San Mateo, California, 1992.

J. Renz and B. Nebel. Qualitative spatial reasoning using constraint calculi. In *Handbook of Spatial Logics*, pages 161–215. 2007.

H. Simon. *The Sciences of the Artificial [The Karl Taylor Compton lectures.]. Chapter 3: The Science of Design*. MIT Press, Cambridge, MA, USA, 1969.

H. Simon. *The sciences of the artificial (3rd ed.)*. MIT Press, Cambridge, MA, USA, 1996. ISBN 0-262-69191-4.

E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, June 2007. ISSN 15708268. doi: 10.1016/j.websem.2007.03.004. URL `http://dx.doi.org/10.1016/j.websem.2007.03.004`.

L. Sullivan. The tall office building artistically considered. *Lippincott's Magazine*, 1896.

M. Thielscher. Introduction to the fluent calculus. *Electron. Trans. Artif. Intell.*, 2:179–192, 1998.

M. Thielscher. Flux: A logic programming method for reasoning agents. *Theory Pract. Log. Program.*, 5(4-5):533–565, 2005. ISSN 1471-0684.

US GSA. US Courts Design Guide, 2007. URL `http://www.gsa.gov/graphics/pbs/Courts_Design_Guide_07.pdf`. Judicial Conference of the United States. US General Services Administration (GSA). April 23 2010.

P. Vermaas, P. Kroes, A. Light, and S. Moore, editors. *Philosophy and Design: From Engineering to Architecture*. Springer.

M. D. Vos. ASP: The future is bright. In *LPNMR*, pages 625–627, 2009.

J. O. Wallgrün, L. Frommberger, D. Wolter, F. Dylla, and C. Freksa. Qualitative spatial representation and reasoning in the sparq-toolbox. In T. Barkowsky, M. Knauff, G. Ligozat, and D. Montello, editors, *Spatial Cognition V: Reasoning, Action, Interaction: International Conference Spatial Cognition 2006*, volume 4387 of *LNCS*, pages 39–58. Springer-Verlag Berlin Heidelberg, 2007.

M. Westphal, S. Woelfl, and Z. Gantner. GQR: A fast solver for binary qualitative constraint networks. In *AAAI Spring Symposium on Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*, 2009.