Computing
DOI 10.1007/s00607-016-0509-6

Architecting dynamic cyber-physical spaces

Christos Tsigkanos[1] · **Timo Kehrer**[1] ·
Carlo Ghezzi[1]

Received: 11 June 2016 / Accepted: 7 July 2016

Abstract We increasingly live in cyber-physical spaces: spaces that are both physical and digital, and where the two aspects are intertwined. Cyber-physical spaces may exhibit a range of behaviors, from smart control of heating, ventilation, and light to visionary multi-functional living spaces that can be spatially re-organized in a dynamic way. In contrast to traditional physical environments, cyber-physical spaces often exhibit dynamic behaviors: they can change over time and react to changes occurring in space. Current design of spaces, however, does not normally accommodate the cyber aspects of modern spatial environments and does not capture their dynamic behavior. Spatial design, although done with CAD tools and following certain international processes and standards, such as Building Information Modelling (BIM), largely produces syntactic descriptions of spaces which lack dynamic semantics. As a consequence, designs cannot be automatically (and formally) analyzed with respect to various requirements emerging from dynamic cyber-physical spaces; safety, security or reliability requirements being typical examples of this. This paper will show an avenue for research which can be characterized as rethinking the design of spatial environments, i.e., dynamic cyber-physical spaces, from a software engineering perspective. We outline our approach where formally analyzable models may be automatically extracted from BIM depending on the analysis required, and then checked against formally specified requirements, both regarding static and dynamic properties of the design, prior to the construction phase (at design time). To realize automated operational management, these models can also be used during operation to continuously check satisfaction of the requirements when changes occur, and possibly enforce their satisfaction through self-adaptive strategies (at run-time).

✉ Carlo Ghezzi
carlo.ghezzi@polimi.it

[1] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy

Keywords Cyber-physical systems · Cyber-physical spaces · Building information modeling · Static and dynamic semantics · Formal verification · Self-adaptation

Mathematics Subject Classification 68N30 Mathematical aspects of software engineering (specification, verification, metrics, requirements, etc.)

1 Introduction

Innovations in various technological fields, such as embedded computing, sensing and communication infrastructures, have led to a cyber-physical world where the boundary between the physical and the cyber world gets increasingly blurry. This phenomenon is reflected by the notion of a cyber-physical system (CPS), a software-intensive system where computational elements heavily interact with physical entities, thus controlling individual, organizational or mechanical processes through the use of information and communication technology [4,48]. A cyber-physical space (CPSp) is a special case of a CPS indicating a spatial environment which includes both cyber and physical elements, i.e., computational and communication features being embedded in physical spaces [58].

Cyber-physical spaces, such as smart buildings, are becoming ubiquitous. Thus, the design of spatial environments, a task which has formerly been exclusively concerned with architecting physical spaces, becomes increasingly challenging. A first important category of new challenges results from the interplay between cyber and physical elements, especially w.r.t. safety, security or reliability requirements [19,40,53,59]. Concerning security threats, for instance, cyber-enabled physical attacks can occur when access control systems protecting assets are cyber-controlled. Conversely, physically enabled cyber attacks can occur when physical access to assets such as computers or networks enables cyber attacks [59]. Moreover, modern cyber-physical spaces are much more dynamic than traditional spatial environments used to be. People or robots moving around connecting and disconnecting from wireless networks are an example of entities dynamically performing actions. Such dynamics have to be considered in the design of spatial environments, e.g., to guarantee adequate response times of medical personnel in a digitally connected hospital environment. Likewise, the increasing need for flexibility, e.g., in densely populated urban areas, will heavily influence the shape of prospective spatial environments. In the CityHome project,[1] for instance, researchers from MIT Media Lab's "Changing Places" group developed a prototype of a highly dynamic studio apartment [39]. It comprises a set of home essentials such as a bed, a work space, a dining room table, a cooking range, and a multipurpose storage occupying minimal physical space. The usable space is maximized by automatically re-arranging the room according to current needs using gestures and voice control. Thus, walls, doors and other building elements are no longer static entities but can be dynamically re-located in the physical space. Finally, like for any other software-intensive system, maintaining a CPSp which "operates" in a dynamic environment is faced with the manifold challenges of software system evolution [8,25,42,63] and

---

[1] http://cp.media.mit.edu/cityhome.

demands for operational management to observe evolution and potentially react to environmental changes.

The current practice of designing physical spaces, however, is weak in facing these emerging challenges [58,59,67]. Computer-aided design (CAD) tools as used by the construction industry largely produce space plans as blueprints for construction. Although digitally accessible in machine readable formats and following certain international standards like Building Information Modelling (BIM), space plans often still serve the role of a merely static documentation purpose. The resulting models are syntactic descriptions of spaces lacking dynamic semantics. In addition, design documents are mostly disconnected from the computational components enabling smart functionalities, a great concern especially in safety-critical spaces such as industrial plants or medical environments. As a consequence, designs cannot be automatically (and formally) analyzed with respect to the various requirements emerging for dynamic cyber-physical spaces. Moreover, "run-time support" to recognize, handle and manage topological changes, essentially some form of automated operational management, is largely missing. The current practice mostly revolves around static compliance checks with respect to norms that are domain specific. The inspection for regulations compliance is often done manually or through check-lists. Some kinds of automated reasoning about static properties of building designs are slowly becoming available, but they are still considered as advanced practices. Automated dynamic analysis and simulation tools are only offered for a few dedicated scenarios such as building evacuation. Each of these scenarios is addressed by dedicated and mostly proprietary modeling and analysis tools. This hinders the interoperability between these tools and thus the development of sophisticated tool chains for systematic model-based engineering of cyber-physical spaces.

In this paper, we present our vision of architecting dynamic cyber-physical spaces by rethinking design and operation of spatial environments from a software engineering perspective. In essence, we argue that a CPSp brings many challenges which arise in a similar way in classical software engineering, especially when considering requirements such as security, safety, reliability or robustness. To mitigate these challenges, the software engineering literature provides a substantial body of knowledge, particularly regarding the analysis of complex software systems using formal verification and validation techniques such as model checking [5,14], or the ability of systems to self-adapt at run-time, reacting to environmental changes [18,51]. Thus, it is a natural and promising approach to adopt software engineering principles for the design and operation of dependable and adaptable cyber-physical spaces. We aim at providing a holistic approach to modeling, analysis, and operation of cyber-physical spaces. From an engineering perspective, we follow basic principles from the field of Model-Driven Engineering (MDE) [11,56]. We ground the representation of analyzable models extracted from BIM on the Essential Meta-Object Facility (EMOF) standard defined by the Object Management Group (OMG) [26]. The extraction itself shall be implemented as a parametric and thus customizable model-to-model transformation [16].

The contributions over the state of the art can be summarized as follows: Traditional space plans serving as syntactical descriptions of physical spaces are enriched by domain-specific physical, cyber-physical and cyber entities of interest. This includes the modeling of the dynamic actions that may be performed by these or any of the traditional types of entities. Syntactic descriptions are translated into analyzable models with well-defined semantics. By providing formal static and dynamic semantics in terms of topological concepts of locality and connectivity of entities it is possible to support many forms of advanced analyses typically performed in software engineering. In particular, dynamic semantics can deal with the dynamism that a space exhibits. It enables design-time formal modeling of possible topological changes that may occur. It also enables formal analysis of their possible effect on quality requirements, hence supporting design-time exploration of different design alternatives, with the goal of optimizing satisfaction of possibly interdependent requirements. Moreover, when the cyber-physical space becomes operational, dynamic semantics allows the space to exhibit autonomous, self-adaptive behaviors. Data gathered by spatial monitors and indicating changes can generate reactions that try to automatically satisfy the requirements. Motivated by the spatial environment of a smart hospital, we show how both qualitative and quantitative properties can be specified and verified at design time, and how the analysis results can be exploited to trigger self-adaptation at run-time. Finally, from an engineering point of view, grounding our approach on MDE principles and standards facilitates the development of integrated and open tool environments for systematic model-based engineering and operation of cyber-physical spaces on top of traditional BIM.

The remainder of the paper is structured as follows. Section 2 briefly reviews the current state of practice in the construction industry and analyzes its shortcomings w.r.t. to emerging challenges arising from the shift to cyber-physical spaces. To further illustrate major challenges more lively and to motivate our vision of rethinking the design of spatial environments from a software engineering perspective, Sect. 3 sketches a requirement analysis of an example smart hospital environment and identifies the basic software engineering principles which may be used for designing dependable cyber-physical spaces. Section 4 presents our first steps towards realizing this vision and provides an overview of our holistic approach to modeling, analysis, and operation of cyber-physical spaces. Section 5 discusses conceptual modeling of cyber-physical spaces and Sect. 6 illustrates how analyzable models can be obtained; they are used in Sects. 7 and 8 to enable design-time and run-time analyses, respectively. Related work will be discussed in Sect. 9 and Sect. 10 concludes the paper along with an outlook on future work.

2 Designing spatial environments

In this section, we briefly review the current state of practice of how the design of spatial environments is supported by information technology and how building designs are checked for compliance against requirements and regulations. Thereupon, we discuss the major challenges emerging for the design and operation of dynamic cyber-physical spaces, many of which being insufficiently addressed by current practices of the construction industry.

2.1 State of practice

The current state of practice in the design of spatial environments in the Architecture–Engineering–Construction (AEC) industry revolves around supporting the specification of physical layouts and structural elements. Physical layouts of spatial environments are typically modelled using a computer-aided design (CAD) software environment. These tools, most of them being commercial solutions such as Graphisoft's ARCHICAD[2] and Autodesk's REVIT[3] to mention just two, produce space plans as traditional blueprints for construction. Using a CAD tool, a designer basically specifies how a space is divided into areas as well as relevant structural, geometric and material properties.

Building Information Modeling (BIM) [20] emerged in the construction industry from the need of a shared digital representation as well as a process to be used throughout a building's lifecycle to facilitate design, construction and operation. BIM provides rich representations of structural and functional characteristics of buildings, ranging from physical layouts to electrical installations. BIM workflows are supported by leading CAD tools [17], and the BIM principles are reflected into the Industry Foundation Classes (IFC) [33] format, which has become the de-facto standard to exchange BIM models. It aims to be interoperable across individual, discipline-specific applications, supporting planning, design, construction, operation and maintenance of physical spaces. Elements found in an IFC specification include structural elements of a building design such as rooms and walls, their position in the space as well as attributes they may have. Essentially, IFC is the "syntactic" spatial description of a building design.

Very often, building designs need to be checked for compliance against requirements and regulations. Such requirements may come from different regulatory authorities, are often domain specific and concern various aspects of a design. Different domains and different practices dictate diverse requirements; an airport's security requirements differ from the medical regulations to which a hospital must conform to, while both must comply with common criteria, e.g., concerning accessibility. Model-based reasoning techniques have been utilized for studying certain properties and scenarios such as area capacities in queue formations or the behavior of people under emergency situations such as evacuation scenarios [24,27]. However, such methods are tailored to specific designs, and application is left to the engineer in an ad hoc manner. The current practices to support conformance with requirements from standards or regulations is based on rule-based checking, and usually human-driven organizational review processes are defined to guide designers and inspectors to review and assess the design. Beyond human-driven processes, taking advantage of the digital BIM/IFC building representations, automation of rule checking has concentrated on building codes and regulations, and has been integrated in several architectural tools. Such rule-based systems [21] assess building designs according to various static properties, expressed as rules, constraints or conditions.

---

[2] http://www.graphisoft.com/archicad.

[3] http://www.autodesk.com/products/revit-family.

2.2 Problems and emerging challenges

The wide application of current practices across the AEC industry has shown the applicability of rule-based checking for conformance to regulations or building code criteria. However, existing approaches face new challenges originating from the increasing shift of merely static physical spaces to highly dynamic cyber-physical ones. This raises the need for additional expressiveness in checking diverse requirements, particularly including CPSp dynamics, and the need for operational management to ensure requirements satisfaction after system deployment.

Interplay of cyber and physical spaces Digital features are increasingly integrated into the physical world, ranging from digitally controlled access to buildings to more elaborate smart spaces, calling for new paradigms, development methodologies and reasoning methods. In such complex CPSp, computing and communication elements must be modeled and considered together with physical elements, with respect to qualities that the composite system should exhibit.

Complex requirements Space design seldom relies on precisely specified requirements, resulting from systematic requirements elicitation processes. On the other hand, requirements for cyber-physical spaces are characterized by an increase in complexity, due to the mutual relations existing between entities residing in the two spaces. For instance, two people may communicate because they are physically located in the same room, or they may communicate using mobile devices which are connected over some network. Additionally, diversity in concerns that requirements must counter in a CPSp may range from security, safety and reliability as commonly understood in software engineering to traditional ones in physical spaces such as movement of people in buildings to energy consumption. Such complex requirements, call both for mathematical precision in their explicit formulation and in new reasoning methods to provide assurances about their satisfaction.

Dynamics in CPSp Typically, cyber spaces are considered to be highly dynamic in nature. Information-driven change is frequent, e.g., when people connect and communicate through wireless networks or new paths become available for people's movement by digitally locking or unlocking doors. In addition, a physical space might also change along with entities inhabiting it, due to people or assets moving and thus changing the topology of the space. This change ethos must be integrated into the reasoning practice as it may affect the satisfiability of requirements of the cyber-physical space.

The above-mentioned problems are insufficiently addressed by the current practices in the AEC industry. Building Information Models still lack a representation of cyber elements and do not consider the interplay between digital and physical entities. Moreover, models used for reasoning still lack semantic information that may enable significant design-time analysis of complex requirements or that are critical for instrumenting run-time actions in a reactive space context. These considerations lead to emerging challenges in the design and systematic engineering of cyber-physical spaces. First, the cyber dimension has to become an integral part in the design of spatial environments when relevant, and the design of physical spaces must be connected with the cyber components that enable smart functionalities. Second, there is an increasing need for expressing complex requirements on spaces and providing formal assurances about their satisfaction. Finally, to assure satisfaction of requirements at run-time, a new paradigm of adaptation is needed treating change as a first-class entity.

3 Cyber-physical spaces: where software engineering meets architecture

In this section, we motivate our vision of rethinking the design of spatial environments from a software engineering perspective. In particular, we argue that requirements of a CPSp should be analyzed, documented up-front and managed: tasks which are well known from requirements engineering. Section 3.1 introduces a motivating example which revolves around the design of a smart hospital environment: a medical facility which has to satisfy a variety of requirements. Subsequently, following well-established software engineering methods, Sect. 3.2 presents our basic principles for designing dependable cyber-physical spaces.

3.1 Requirements analysis of an example smart hospital environment

The environment of a smart hospital consists of a physical space comprising areas with various functions, along with a cyber space that includes mobile devices, a networking infrastructure and medical data. Agents with various roles such as medical personnel, patients and visitors roam inside the physical space and interact with entities in the cyber space. Figure 1 sketches the CPSp we use as an example, along with various assets and agents. The physical layout is a typical intermediary design of a building architect. Agents such as doctors and patients being located in rooms are part of the physical space. Network connections (represented as dotted lines) corresponding to

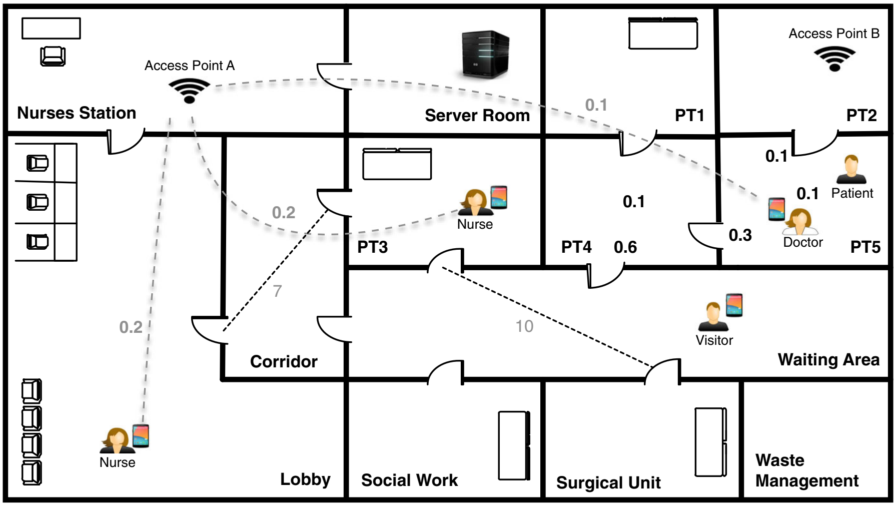

Fig. 1 Sketch of a cyber-physical space of a smart hospital environment

the cyber space are also included in Fig. 1; they traverse various areas, e.g., linking agents to the access point located in the nurse's station. In the context of this paper, construction-specific elements (such as materials and dimensions) are ignored, and a topology-driven approach is adopted; we are interested in relationships inherent in the space, be it between building entities, people or assets. Moreover, we assume the behaviors of agents in the smart hospital environment to be independent of each other.

The design of a smart hospital environment has to satisfy a variety of domain-specific requirements. Hereafter, we motivate and informally present a significant sample. The first case is representative of requirements that deal with compliance with existing regulations. Current hospital regulations require that a Surgical Inpatient Unit (SIU) must be accessible by doctors or patients from Social Work services but it must not be adjacent to Waste Management facilities [61]. We can consider this as a simple (static) safety requirement a hospital space design must satisfy.

> (*R1*) *Safety* A Surgical Inpatient Unit (SIU) must be directly accessible from Social Work services but not adjacent to Waste Management facilities.

A hospital infrastructure serves several medical purposes which require communication and movement of people (i.e., agents) with a variety of roles, such as nurses or doctors tending to patients or emergency personnel providing services. Visitors of admitted patients may also roam inside public areas. As hospital areas have different geometric sizes, different times are required for an agent to traverse each room (e.g., in Fig. 1, a traversal from the corridor to PT3 takes 7 seconds). Moreover, agents (such as doctor and nurses) may carry networked pager devices which allow them to contact each other.

A hospital should exhibit specific properties expressing temporal and spatial concerns revolving around emergency responses. A typical scenario of interest involves a nurse needing to contact a doctor for a patient emergency [45]. In such a case she has two options: either a) contact the doctor through the network with her mobile device, or b) physically locate her inside the hospital. Reliability in a smart hospital emergency setting could entail that a nurse must be always able to reach the doctor (physically or through the network) within a certain timeframe with at least a certain probability. Satisfaction of such a requirement depends on characteristics of both the physical and the cyber space.

Regarding the physical space, its topology is highly relevant. For example, if two agents are located in far-away rooms person-to-person communication is hindered. Additionally, room sizes must also be taken into account; traversal time (time needed to traverse a room from door to door) depends on size. For example, traversal time is higher for a long corridor than for a small room. Traversal times can be estimated from the geometric attributes of the physical space layout. The way agents move inside the hospital also affects the requirement. More precisely, it can be safely assumed that in each room, each agent may stay in a room or move to any door-adjacent room. These actions taking place in the CPSp may be associated with probabilities (shown inside rooms in Fig. 1) which can be, for example, gathered from access logs, domain observations or domain knowledge. To locate the doctor physically, the nurse must exhaustively search rooms until she finds her. Knowing that she is more likely to find the doctor in some hospital areas (e.g., patient rooms), she employs a search strategy, so her moves inside the physical space are also associated with probabilities.

Regarding the cyber space, wireless coverage varies, and this has an effect on communication between agents, as access points have variable range and they must not be placed near sensitive medical equipment. Wireless signal coverage in the space depends on relative positions of communicating agents, information that can be estimated through wireless networks engineering methods [49]. These propagation models take into account building materials, access point location, characteristics and transmission power, and produce success probabilities of connections associated with relative positions of agents in pairs of rooms. Thus, to locate the doctor through the cyber space, the nurse attempts to page the doctor through the network from every room, in a probabilistic manner. Probabilities of successful paging operations by nurses are shown in gray in Fig. 1.

> (*R2*) *Reliability* An attending doctor must be reached by a nurse either physically or through her mobile device with a probability of at least 80 %, within 50 seconds.

Hospitals are commonly faced with scenarios where a critical patient is being transferred across the physical space. Such a scenario for example, may occur when a doctor moves a critical patient across general care facilities to a surgical unit. Presence of non-medical persons nearby without being accompanied by medical staff, poses risks for the critical patient's condition. We can regard this as an integrity requirement that needs to be satisfied by the smart hospital environment.

> (*R3*) *Integrity* No visitors unaccompanied by a nurse are allowed to be in the same room with a doctor when he is with a critical patient.

Wireless networks are key to a smart hospital environment, as they not only facilitate communication between doctors and nurses, but also provide a means to access patient information. Visitors at the hospital also enjoy wireless services, which for the sake of this example are assumed to be of lesser importance. Two wireless networks exist in the cyber space of our example, one providing highly available, fast connectivity and a legacy one, providing a basic level of service to connected clients. Access and storage of patient healthcare data is one of the main functions of a contemporary medical facility; these data are confidential and subject to privacy laws (e.g., US HIPAA [43]). In a typical scenario, a doctor attending to a patient in an examination room accesses a patient's medical history data stored on the hospital server. Such connections should always take advantage of the fast highly available network, while ensuring that information flowing should not be eavesdropped. As such, when a doctor accesses patient information, visitors must be disconnected from this fast network. Regarding the physical aspect, visitors present in the examination room may also violate data confidentiality, by physically eavesdropping sensitive information in a device's screen for instance. We regard this as a confidentiality requirement, spanning both cyber and physical spaces.

> (*R4*) *Confidentiality* To ensure confidentiality of patient data, no visitors must be connected to the same network when a doctor is accessing patient data, or present in the same room as her.

3.2 Designing dependable cyber-physical spaces

The hospital example introduced in the previous section highlights the need for precisely formulating requirements and then assuring dependability of the designed CPSp by showing that all requirements are indeed fulfilled. To provide formal assurances, software engineering principles and methods can be adopted. Those utilize specification through model-based approaches which result in models with well-defined semantics. Such models enable automated analysis of requirements using formal verification techniques.

Such model-based rigorous specification of cyber-physical spaces and the automated analysis of requirements can target both design-time and run-time facets of the CPSp; in the same fashion that DevOps [7] principles are adopted in the traditional software engineering field. We highlight specific features that a *model-centric DevOps for CPSp* approach needs to support.

A holistic, model-driven view of CPSp A conceptual model that integrates both cyber and physical aspects of the space is needed. Both requirements R2 and R4 concerning reliability and confidentiality require reasoning about relationships that span both cyber and physical spaces. For instance, requirement R2 concerns the communication between the doctor and the nurse, either by co-location in the same room (physical space) or through the network (cyber space). As for confidentiality requirement R4, the interplay of cyber and physical spaces is also prominent. A doctor is with a patient (in the physical space) and connected to the patient's data (in the cyber space); at the same time, no visitors must be present in the room (in the physical space) or connected to the same wireless network (in the cyber space).

Static and dynamic analyses of CPSp Static analysis aims at reasoning about latent qualities of a CPSp design. Invariants on topological relationships between entities as found in safety requirement R1 are an example of this. Reasoning about such static properties demands for structural models and a proper formalism to express structural constraints and invariants. Conversely, dynamic analysis concerns the dynamism that a CPSp configuration exhibits when additionally considering the ways it may change, as changes can lead to requirements violations. In fact, satisfaction of requirements R2 and R4 depends on topological changes that occur in the CPSp. For instance, if a visitor connects to the same network a doctor is using to access patient data, R4 is violated. In the same fashion, the collective effect of medical personnel roaming and communicating in the space determines satisfaction or violation of R2.

Design time and run-time The extent of assurances obtained from analysis depends on whether they address concerns that can arise at design time or run-time. When contemplating design-time analysis, the overall objective is that the CPSp must be correct "by design"; changes that may occur in the configuration when the space is deployed cannot violate the requirements, given that their assumptions are met. This is the case with requirements R1 and R2. Regarding R1, positions of medical areas in the physical space or medical equipment that a requirement predicates about can be safely assumed to be fixed. Likewise, assuming the physical layout of the CPSp, wireless coverage and probability distributions of agents to be stable, satisfaction of requirement R2 can be guaranteed by a proper design, too.

On the contrary, satisfaction of requirements R3 and R4 cannot be guaranteed at design time, and reasoning must be offloaded to run-time, i.e., some form of operational management of the CPSp is needed. Concerning requirement R4, for instance, operational management can be responsible for access control by preventing visitors from entering certain rooms, or actively forcing disconnection or handover of visitors from the critical wireless network. The challenge of moving parts of the reasoning and enactment to run-time highlights the need for *adaptation*.

Self-adaptation An adaptive approach is needed to discover possible requirements violations determined by topological changes in the CPSp and then counteract by enacting measures to ensure requirements satisfaction. This can be achieved by keeping a model of the CPSp alive at run-time, monitoring the environment of changes and updating the model which drives adaptation in the form of appropriate measures which counter requirements violations.

4 Towards a holistic view on modeling, analysis, and operation of cyber-physical spaces

Our vision for a holistic approach to modeling, analysis, and operation of cyber-physical spaces is illustrated in Fig. 2. As illustrated in the upper part, the creation of a conceptual model of a CPSp grounds on building information modeling tech-

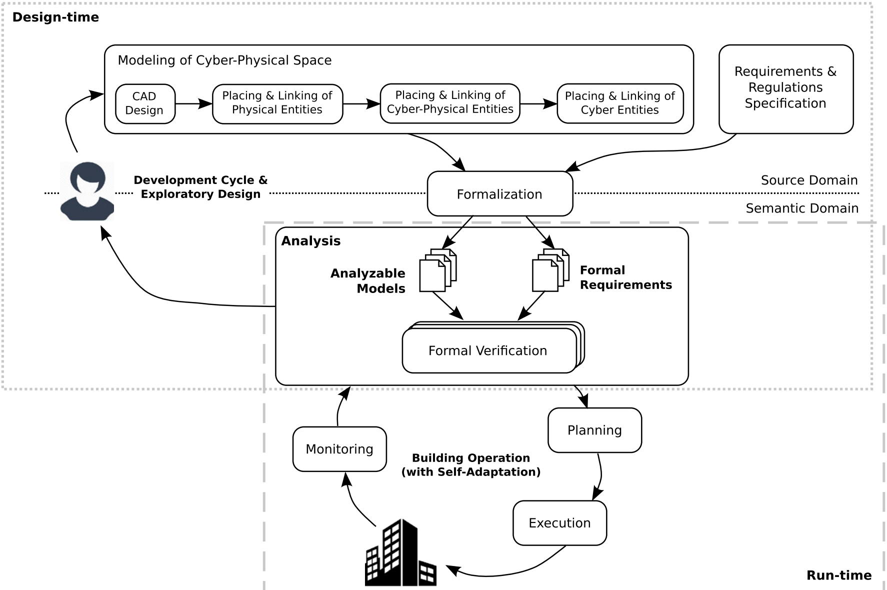

Fig. 2 Overview of the approach

niques used in current practice. Standard building information models are enriched with additional physical, cyber-physical and purely cyber entities being considered as relevant in the context of the CPSp under design. Section 5 discusses the integration of cyber aspects with traditional spatial modeling and outlines a possible process to enrich standard BIMs to become conceptual models of a CPSp.

Subsequently, the conceptual model of a CPSp is translated into formal models enjoying well-defined static and dynamic semantics. We chose bigraphs and Bigraphical Reactive Systems [44] as formal notations to model both static topological concepts, such as locality and connectivity of entities, and dynamic changes of the CPSp over time (see Sect. 6). This formalization aims to encode a conceptual model into a form that facilitates various kinds of automated reasoning. Different analyzable models may be automatically generated to support different kinds of analyses, as discussed in Sect. 3.1 for the smart hospital running example.

In addition to the translation of the conceptual model of a CPSp into analyzable models, the use of advanced formal verification techniques requires a second step, namely the formal specification of requirements derived from the source domain, which are assumed to be documented up-front, e.g., in documents standardized by regulatory authorities. These requirements are formalized into properties to facilitate automated reasoning.

The use of the results of the various analysis procedures being enabled by our approach is twofold, targeting both design time and run-time. At *design time*, positive analysis results provide *formal assurances*, e.g., regarding the compliance with standard regulations. Negative results may provide valuable feedback to architects and enable *exploratory design*. Some of the possible design-time analyses, including a formalization and verification of two of the sample requirements introduced in Sect. 3.1, will be illustrated in Sect. 7. At *run-time*, i.e., when a CPSp is in operation, advanced analysis techniques are an essential prerequisite to ensure that possible changes occurring in spaces, for example, due to actions performed by agents, do not lead to violations of requirements. This feature makes the CPSp *self-adaptive*. As shown in Fig. 2, the results of analyzing models at run-time, which are continuously updated through monitoring the CPSp in operation, trigger the planning and finally execution of the self-adaptive loop, through which the CPSp reacts to achieve requirements satisfaction. In Sect. 8, we sketch potential benefits of run-time reasoning and self-adaptation addressing the sample requirements R3 and R4 introduced in Sect. 3.1.

5 Conceptual modeling of cyber-physical spaces

A conceptual model of a CPSp is an abstraction of a spatial environment which, in contrast to standard BIM, covers both the physical and the cyber dimension of a CPSp, along with additional data being required for the intended analyses (e.g., traversal times between doors of rooms). Analogous to BIM, such a conceptual CPSp model is a purely syntactical description which we assume to be defined using a "domain-specific modeling language (DSML)" [62], or a set of DSML dialects, which enables CPSp designers such as building architects and construction engineers to express a design in terms of well-known concepts of the given domain. As shown in the upper part of Fig. 2, the conceptual modeling of cyber-physical spaces starts with designing the physical space using state-of-the-art CAD tools. Subsequently, the traditional space plan obtained in this first phase is enriched with additional physical, cyber-physical and purely cyber entities being relevant in the context of the CPSp under construction. We finally obtain a model which, following common terminology of model-based software engineering, can be considered as a multi-view model [22], which covers both the physical and the cyber dimension of a spatial environment and allows us to consider a CPSp from different viewpoints:

BIM view The *BIM view* of a conceptual model of a CPSp reflects the traditional notion of a space plan of a spatial environment as produced using standard CAD tools.

Physical entities view The *physical entities view* comprises physical entities which are not part of traditional space plans. Agents, such as the nurse and the doctor in our smart hospital, are examples of such entities, which are located in the physical space.

Cyber-physical entities view The *cyber-physical entities view* comprises special physical entities representing gateways to the cyber dimension. Wireless network access points, PC workstations, mobile phones and other devices are typical examples of cyber-physical entities. Just like additional physical entities, cyber-physical entities are located in the physical space.

Cyber entities view The *cyber entities view* models entities which are purely digital but which are relevant for certain requirements analysis. Files containing sensitive patient data are an example of cyber entities being of interest for our hospital example. Typically, cyber entities are located at and linked through cyber-physical entities serving as interfaces to the digital world.

Conceptual CPSp meta-model As usual in model-based engineering, we assume the types and attributes of conceptual CPSp elements and the possible relationships between them to be defined by a meta-model, essentially the abstract syntax definition of the underlying DSML. As illustrated in Fig. 3, a conceptual CPSp meta-model comprises the definition of four sub-models corresponding to the viewpoints introduced above. The Building Information Meta-model corresponds to a standard BIM meta-model, such as IFC, and may define extensions such as further attributes of entities, e.g., to capture additional information required for the intended analyses (see Sects. 7 and 8). Moreover, the conceptual CPSp meta-model comprises the definition of phys-

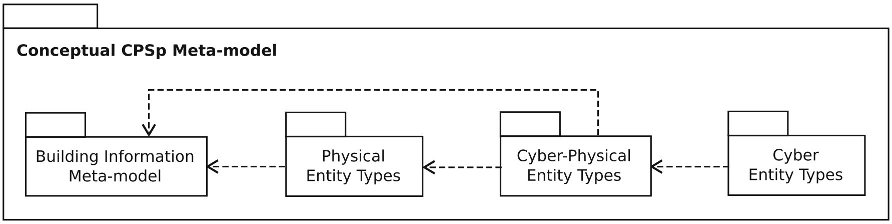

Fig. 3 Conceptual meta-model for modeling cyber-physical spaces

ical, cyber-physical and cyber entity types; this includes the definition of attributes, the possible relationships among entities of these types as well as how entities can be structurally integrated into the model of the physical space.

Tooling and implementation To integrate additional entities into standard BIM models, our approach is open to any BIM modeling approach and possible tool chain. One possible implementation of the conceptual CPSp meta-model of Fig. 3 is proposed in [58]. In this approach, which can be considered as a tight coupling approach, we extend BIM/IFC [33] by additional entity types such as *Agent*, *PhysicalEntity*, *ComputingDevice*, and *CyberEntity*. These entity types are defined as subclasses of the IFC meta-class *IfcProduct*. Another tight coupling approach based on BIM/IFC and the CAD modeling tool Revit has been demonstrated in [67]. Here, IFC shared parameters, family property parameters and the mark tag are used to extending BIM/IFC for different kinds of sensors and actuators (see Sect. 9).

An alternative approach is to integrate additional entities into standard BIM models in a loosely coupled manner. This way, standard BIM modeling tools are not extended but rather complemented by separate tools. Model interrelations are given by, e.g., equal names of model elements. A detailed discussion of such implementation-related issues is out of the scope of this paper. Concrete design decisions heavily depend on the standard BIM tools being integrated into the overall tool chain and how these tools can be extended by additional concepts or plug-ins.

6 Towards analyzable models

To add a precise meaning to syntactical descriptions of a CPSp, we choose a translational approach [13] mapping the conceptual model (or a subset of the conceptual model) to formal models with well-defined static and dynamic semantics. Due to their inherent concepts of locality and linking, we choose bigraphs [44], essentially a process meta-calculus to embed and thus unify reasoning over a set of existing formalisms and calculi, as the semantic domain of our translational CPSp semantics. This enables us to infer and express the topology of a CPSp using bigraphs in a natural way. Thereupon, dynamic reconfigurations, i.e., the possible ways of how a CPSp may change over time, can be modeled as a Bigraphical Reactive System (BRS) [44].

In the remainder of this section, we briefly introduce bigraphs as our semantic domain and illustrate topological modeling (providing static semantics) and modeling of reconfigurations (providing dynamic semantics) using the smart hospital example of Sect. 3.1. We finally discuss several aspects of an engineering solution supporting the (semi-)automated translation of a conceptual model of a CPSp into analyzable models with well-defined semantics.

6.1 Bigraphs as the semantic domain

Bigraphs have been proposed by Milner [44] as a fundamental theory and modeling formalism for structures in ubiquitous computing [65]. The anatomy of bigraphs is based on two fundamental concepts of discrete spaces: locality and connectivity, called *placing* and *linking* in bigraphical terminology. In essence, a bigraph consists of a *place graph*, a forest defined over a set of nodes which is intended to represent entities and their locality in terms of a containment structure, and a *link graph*, a hypergraph composed over the same set of nodes representing arbitrary linking among those entities. Connections of an edge with its nodes are called *ports*. Place and link graphs are orthogonal, and edges between nodes can cross locality boundaries. The types of nodes, called *controls* in bigraphical terminology, are defined by a so-called *signature*.

What follows is an informal introduction to bigraphs by an inductive definition of an algebraic notation for bigraphs; the interested reader is referred to [44] for complete formal treatment.

$$P.Q \qquad Nesting\ (P\ contains\ Q) \tag{1a}$$

$$P \mid Q \qquad Juxtaposition\ of\ nodes \tag{1b}$$

$$-_i \qquad Site\ numbered\ i \tag{1c}$$

$$K_w.(U) \qquad Node\ with\ control\ K\ having\ ports\ with\ names\ in\ w.\ K\ contains\ U \tag{1d}$$

$$W \parallel R \qquad Juxtaposition\ of\ bigraphs. \tag{1e}$$

As shown by Formulae 1a–1e, bigraphs can be described through concise algebraic expressions in a process calculus fashion. The containment relationship, i.e., hierarchical nesting of nodes, is expressed in Formula 1a, while juxtaposition, i.e., the placing of nodes on the same hierarchical level, is captured by Formula 1b. Moreover, bigraphs can contain *sites* (Formula 1c) that can be used to denote placeholders, i.e., they indicate the potential presence of further unspecified nodes. Controls are names that define a node's type; each node control can be associated with a number of named ports. In Formulae 1a–1e, P, Q, and U are controls of bigraph nodes. If a single instance node of that type exists in the bigraph, the control also uniquely identifies that node. Otherwise, port names are used as a way to uniquely identify nodes. In Formula 1d, the node identified by control K and port name w also contains U. Ports that appear in a formula with the same name are connected, forming a hyperedge with that name, called *link* in the sequel. Bigraphs can be contained in roots that delimit different hierarchical structures; in Formula 1e, W and R are different roots.

The set of available controls can be defined up-front by a *signature* (not shown in Formulae 1a–1e). A signature is basically a set of pairs of the form $control : arity$, with $arity$ being a natural number defining the number of ports of a node of type $control$. In the following, we abstain from explicit definitions of bigraphical signatures for our example.

6.2 Static semantics: inferring the topology of a CPSp

Our objective is to express topological information inherent in a conceptual CPSp model through locality and connectivity relations, mapping it to a bigraph placing and linking structure. Standard BIM entities as well as additional physical, cyber-physical and cyber entities are mapped to bigraph nodes. Entity types such as *Door*, *Wall*, and

Server are used to identify node controls, while the entity name corresponds to a port name uniquely identifying it. Intuitively, the placing structure of a building floor, for instance, is obtained by juxtaposing all the rooms of this floor and subsequently nesting in each room nodes corresponding to entities contained in that room. For a wall, more than one node is created; each is nested inside nodes representing the rooms bounded by the wall. Similarly, two nodes are created for each door; these are nested inside the nodes representing the rooms connected by the door. For example, the smart hospital of Fig. 1 is represented as a juxtaposition of rooms, as partially shown in Formula 2. Here, the nurses station, the server room and the surgical unit are shown, represented by nodes $Room_{nrs}$, $Room_{srv}$ and $Room_{surg}$, respectively. We abstract away other rooms using a site ($-_4$) juxtaposed to these room nodes. The $Server$, a cyber-physical entity, is located (i.e., contained) in the server room.

$$Room_{nrs}.(-_0) \mid Room_{srv}.(Server.(-_1) \mid -_2) \mid Room_{surg}.(-_3) \mid -_4 \qquad (2)$$

To populate the linking structure, connectivity relations in the space are obtained; they can be either physical or digital. In the physical space, connectivity refers to adjacency relations of physical entities. For example, to connect a room to another via a door, a $Door$ node is placed in the corresponding $Room$. The port of this $Door$ is then linked to the respective $Door$ node contained in the $Room$ the door leads to. Similarly, rooms can be connected by walls. In Formula 3, for example, the Surgical Unit ($Room_{surg}$) is connected to the Social Work area ($Room_{soc}$) through $Wall_q$.

$$\begin{aligned} &Room_{nrs}.(AP_{wlan} \mid -_0) \mid Room_{lobby}.(Nurse.(Pager_{wlan}) \mid -_1) \\ &\quad \mid Room_{surg}.(Door_x \mid Wall_q \mid -_2) \mid Room_{soc}.(Wall_q \mid -_3) \mid -_4 \end{aligned} \qquad (3)$$

Just like standard BIM and other (cyber-)physical entities, cyber entities are treated in the same way in the bigraphical representation, using the same notions of containment and connectivity. For instance, a server (a cyber-physical entity) being placed in the physical space may contain a cyber entity, e.g., a file representing patient's information. Logical connections between entities in the cyber space also have a correspondence in the bigraphical linking structure. This may, e.g., refer to wireless signals forming networks. In Formula 3, for instance, the ports named $wlan$ link the access point AP and the nurse's $Pager$.

Finally, attributes of conceptual CPSp entities are also mapped to the bigraphical representation. The general procedure for the treatment of attributes is to create an $Attributes$ node inside a node. Such an $Attributes$ node serves as a container where attribute keys are represented as inner nodes; each of them is linked to a name representing the attribute value. In cases where attributes are not of interest they can be abstracted by sites, as for example in Formulaes 2 and 3.

6.3 Dynamic semantics: CPSp reconfigurations

Having defined how bigraphs provide topology-driven static semantics of cyber-physical spaces, we proceed to consider how these spaces may change, thus giving rise

$$Nurse.(Dev_{wlan}) \parallel AP_{wlan} \parallel Doctor.(Dev_{wlan}) \rightarrow$$
$$Nurse.(Dev_{wlan}) \parallel AP_{wlan} \parallel Doctor.(Dev_{wlan}.(PNG))$$

$connect$ allows a $Visitor$'s device $wlan$ name to be linked to the one of the access point AP.

$$AP_{wlan} \parallel Visitor.(Dev) \rightarrow AP_{wlan} \parallel Visitor.(Dev_{wlan})$$

$move\_patient$ moves a $Doctor$ who is co-located with a $Patient$ in a $Room$ (r) to another $Room$ (v) which connected by a $Door$ (x).

$$Room_r.(Doctor.(-_0) \mid Patient \mid Door_x \mid -_1) \mid Room_v.(Door_x \mid -_2) \rightarrow$$
$$Room_r.(Door_x \mid -_1) \mid Room_v.(Door_x \mid Doctor.(-_0) \mid Patient \mid -_2)$$

$patient\_data$ refers to a $Doctor$, who when connected to the access point through her mobile device, may access patient data located in the $Server$.

$$AP_{wlan} \parallel Server.(PData) \parallel Doctor.(Dev_{wlan}) \mid \rightarrow$$
$$AP_{wlan} \parallel Server.(PData_{lnk}) \parallel Doctor.(Dev_{wlan,lnk})$$

Fig. 4 Partial BRS specification of the hospital example

to dynamic behavior. This is formalized by Bigraphical Reactive Systems (BRS) [44], which extend bigraphs by adding reaction rules defining possible reconfigurations. Reaction rules are parametric and specify how a bigraph can be modified by selectively rewriting some of its portions. Reaction rules have the general form of $R \rightarrow R'$, where R and R' are bigraphs called *redex* and *reactum*, respectively. If a part of a bigraph that matches the redex is identified, it can be replaced with the reactum, in a fashion similar to graph rewriting. A BRS allows us to describe possible ways in which cyber and physical spaces can evolve through reaction rules. For instance, a fundamental reaction from the scenario presented in Sect. 3.1 is the ability for a doctor to enter a room in the hospital, when she is located next to a door leading to it:

$$\begin{aligned}&Room_r.(Doctor.(-_0) \mid Door_x \mid -_1) \mid Room_v.(Door_x \mid -_2)\\ &\quad\rightarrow Room_r.(Door_x \mid -_1) \mid Room_v.(Door_x \mid Doctor.(-_0) \mid -_2)\end{aligned} \tag{4}$$

As Formula 4 illustrates, utilizing the parameter matching facilities of the formalism through sites, the $Doctor$ moves into $Room_v$, while other entities contained in the $Doctor$ (such as her pager) or the adjacent $Room_r$ are not modified. In the same fashion, we can specify further reaction rules. Essentially, using the reaction mechanism, the designer provides elementary reconfigurations reflecting change primitives required for the desired analyses. Reconfigurations can include, for instance, people moving inside the physical space or establishing connections between devices interacting in the cyber space. A partial specification of a BRS of the hospital example is shown in Fig. 4, where variables r, x, v appearing in formulae range over names.

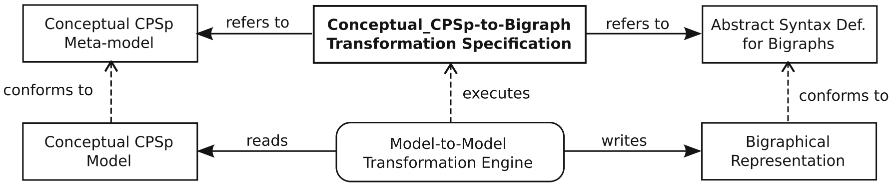

Fig. 5 Bigraphical formalization as a model-to-model transformation

6.4 Towards an engineering solution

We illustrated how the static structure and the dynamic evolution of a CPSp can be formally modeled using bigraphs and bigraphical reaction rules. Our final goal is to provide a proper engineering solution supporting CPSp designers in doing these tasks. Moreover, instead of having a chunk of different analysis tools using proprietary models extracted from BIM, our goal is to develop an open and interoperable tool environment supporting the integration of various analyses on top of traditional BIM, striving for systematic model-based engineering and operation of cyber-physical spaces.

Topology inference as model-to-model transformation As illustrated in Fig. 5, the bigraphical formalization of static CPSp model structure can be conceptually considered as an exogenous model-to-model transformation [16]. It bares the potential to be fully automated by a model transformation engine, taking a conceptual CPSp model as input and producing the bigraphical model as output. The transformation specification is the main parameter passed to the transformation engine. It implements the transformation rules and a rule-scheduling strategy. The conceptual CPSp meta-model serves as the source meta-model of the transformation. An abstract syntax definition for bigraphs, as presented in [36], represents the target meta-model over which the transformation specification is being defined.

The model-to-model transformation of Fig. 5 can be implemented in a step-wise manner. In first step, we intend to extract the topology from standard BIM models using existing work, e.g., the approach presented in [38] (see Sect. 9). In a second step, the obtained graph can be transformed into a bigraph which is enriched by additional entities from the conceptual model. Besides the transformation specification, the behavior of the transformation engine can be influenced by further configuration parameters (not shown in Fig. 5), e.g., to reduce the scope of the transformation to a particular subset of the conceptual CPSp model being relevant in a certain context.

Supporting the modeling of CPSp reconfigurations The engineering solution we aim at to support designers in modeling CPSp reconfigurations is twofold and adopts recent advances from the field of model-based software engineering. On the one hand, we plan to generate reaction rules modeling change primitives, e.g., an agent entering a room, from the conceptual meta-model, following the approach presented in [35]. On the other hand, concerning more complex reconfigurations which are highly application specific, e.g., the handover of agents' wireless connections between different networks, our objective is to enable CPSp designers to specify possible changes in their standard architectural tool environment. To this end, we follow the principle of model transformation by example [34]. The idea is to infer complex reconfiguration rules from examples specified by domain experts using their standard tools and DSMLs. A first rather simple approach is to simply translate the pre- and post-state of an example, i.e., the states of a conceptual model before and after a reconfiguration takes place, to bigraphs representing the redex and the reactum of the corresponding reaction rule. This approach, however, relies on manual postprocessing since sites serving as placeholders have to be added to make reaction rules parametric. Moreover, the obtained reaction rules may have to be reduced to the essential context which is required for a particular reaction taking place. A more advanced solution is to adopt a learning approach and to infer a reaction rule from a set of examples. Each example serves as a concrete reconfiguration instance, the set of which shall be generalized to reconfiguration rules using inference techniques as presented in [3].

Integration with Model-Driven Engineering As shown later in Sects. 7 and 8, bigraphs and BRS obtained from conceptual CPSp models may be translated to other modeling formalisms, depending on the kind of analysis that shall be supported. Typical examples of this are state-transition-oriented models supporting various forms of model checking [5, 14]. In fact, our representation of bigraphs and BRS is based on the abstract syntax presented in [36], which is compliant with the Essential MOF (EMOF) standard defined by the Object Management Group (OMG) [26]. This facilitates the integration with mainstream technologies for Model-Driven Engineering (MDE) [11,56], typically based on the EMOF standard.

7 Design-time analysis and verification

Design of a CPSp must guarantee conformance to the requirements. Static analysis at design time can be achieved by taking advantage of the previously presented static semantics, with requirements represented as bigraphical matching properties expressing configurations of interest. Dynamic analysis at design time can instead be enabled through model checking.

7.1 Static analysis

Static analysis concerns predicating over the static structure that a CPSp exhibits. Although for physical spaces this is an established practice in the AEC industry achieved through rule-based checking [21], here we discuss an alternative pattern-based method evaluated through graph matching over a CPSp configuration.

A static property of a given cyber-physical space can be expressed as a bigraph pattern, describing certain relations of connectivity and containment of entities, in a parametric way. A configuration described by a bigraph pattern satisfies a property if the bigraph specifying the property can be matched against it, meaning that it exhibits containment and connectivity relations among entities as desired. Failure of matching the bigraph representing the property means instead that the property is not satisfied. The utilization of sites in the bigraph specifying the property indicates that the portion of the configuration that matches a site does not affect satisfaction. For example, given that variables x and q range over names, utilizing boolean connectives and elementary predicates expressed in terms of bigraphs, the property which formally specifies the safety requirement R1 has the following form:

$$\begin{aligned}\mathbf{R1} : Room_{surg}.(-_0) \Rightarrow {} & \big(Room_{surg}.(Door_q \mid -_1) \mid Room_{soc}.(Door_q \mid -_2)\big) \\ & \wedge \neg\big(Room_{surg}.(Wall_x \mid -_3) \mid Room_{waste}.(Wall_x \mid -_4)\big)\end{aligned} \tag{5}$$

Formula 5 states that should a surgical unit $Room_{surg}$ exist in the model under consideration, it must not share a wall with any $Room$ with name $waste$; however, it should be connected through a $Door$ to a social work area $Room_{soc}$. Due to the presence of sites in the property specification, other entities that may be contained in rooms do not affect satisfaction. Satisfaction of such a property is checked automatically through bigraph matching [10,44]. For the example of Fig. 1 matching will fail, as there is a wall connecting the surgical room with the waste management room; the designer must re-arrange the space.

7.2 Dynamic analysis

In this section, we discuss how reliability requirement R2 can be verified through dynamic analysis.

Having obtained a BRS describing the dynamics of a CPSp (as illustrated in Sect. 6 and in Sect. 3.1 for the hospital example) along with a bigraph describing the configuration of the space, a wide range of dynamic analyses can be performed. In this section, we discuss how to perform analysis by first interpreting the BRS over some form of a Labelled Transition System [14], an analyzable model (see Fig. 2) describing the CPSp and its evolution in terms of states and transitions. In such a model, states specify configurations of the system, while transitions describe how configurations can change by moving from states to their successors. Given a bigraph that describes the initial configuration, the system evolves by applying reaction rules, which model the occurrence of possible actions in the CPSp, generating new configurations. At each step, several applications of reaction rules may be possible, thus branching off new possible configurations. In our specific running example, hereafter we show that transitions are associated with probabilities and rewards, and this enables reasoning with a probabilistic branching temporal logic.

To verify requirement R2, we must consider the dynamic behavior of agents, which is modelled by reactions defined in the BRS. Recall that the moves of agents inside the hospital are associated with probabilities, reflecting the likelihood that an agent enters a specific room from a another. Moves are also associated with rewards, i.e., numerical values representing traversal times. Intuitively, an agent's behavior can be captured by a Discrete-Time Markov Chain [14] (DTMC), a discrete-time transition system with discrete probability distributions and rewards. In Fig. 6a, a DTMC partially represents the doctor's behavior, where probabilities and rewards are indicated by gray labels on transitions. For example, if the doctor is in room PT5 (state a), she may either stay inside with probability 0.1, enter the patient room PT4 (state b) with probability 0.3, move the patient to room PT4 (probability 0.3) or to room PT2 (probability 0.3). For

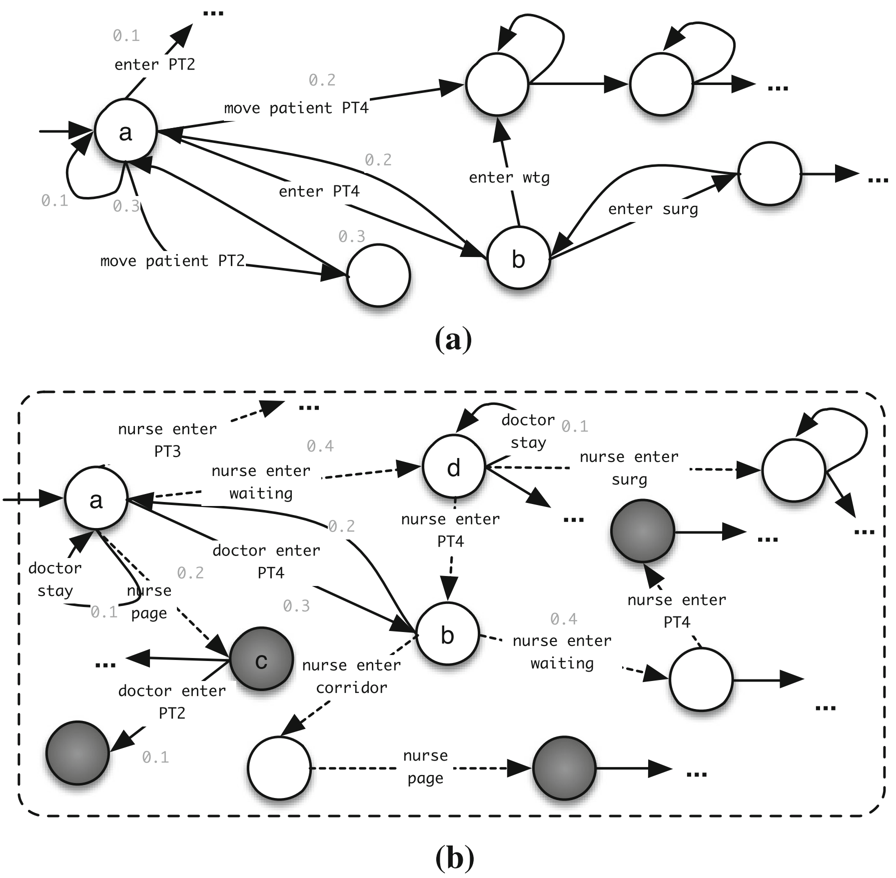

Fig. 6 Design-time dynamic analysis of R2

each state of the DTMC, atomic propositions label the state, declaratively representing the bigraphical configuration of the state; state a in Fig. 6a represents the bigraphical configuration of Fig. 1. The configuration evolves as the doctor probabilistically moves inside the physical space.

Subsequently, we consider the collective behavior of the agents in the CPSp that can be conceived as a system in which processes operate concurrently and asynchronously. The overall model of the system will be a parallel composition of DTMC models representing the behavior of individual agents, reflecting the fact that agents may freely perform actions (from the ones available to them) at any time. This will introduce non-deterministic *choices* in the model, yielding a Markov Decision Process (MDP) [29] with rewards [23]. The probabilistic distributions that describe behavior of individual agents are independent. Non-determinism arises in a state when two different agents concurrently perform probabilistic (atomic) actions and independently change their states. The overall behavior of the system will be defined by the concurrent execution of all agents and captured by the MDP; in each state, a non-deterministic *choice* occurs between several discrete probability distributions of agent's moves to successor states.

Figure 6b shows a fragment of the MDP generated from the parallel composition of DTMCs corresponding to the doctor's and nurses moves (dotted transitions) inside the CPSp; state *a* represents the bigraphical configuration of Fig. 1 as the initial state, while state b represents the configuration where the doctor moved to PT4 while the nurses stayed in PT3 and lobby area. State c corresponds to a configuration resulting from a successful paging operation by a nurse, while state d where a nurse entered the waiting area. Note that as bigraphical predicates encode configurations in each state, states labeled b in Fig. 6a, b represent the same configuration.

The MDP formalism enables automated analysis of a wide range of quantitative properties specified through a probabilistic temporal logic. Probabilistic Computation Tree Logic (PCTL) [28] is such a branching time logic which extends CTL [14] with a probabilistic operator (`P`), manifested as quantitative extensions of CTL's *all* (`A`) and *exists* (`E`) operators. Moreover, PCTL may be supplemented with consideration of rewards (operator `R`). Model checking for PCTL involves determining states of an MDP satisfying a PCTL formula.

$$\begin{aligned}\mathbf{R2} : {} & [\mathtt{R}_{\leq 50}\mathtt{P}_{\geq 0.8}\mathtt{F}\ Nurse.(-_0)\big|Doctor.(-_1)] \\ & \vee [\mathtt{P}_{\geq 0.6}\ \mathtt{F}\ Doctor.(Dev_{wlan}.(PNG))]\end{aligned} \tag{6}$$

Formula 6 specifies reliability requirement R2, to be evaluated over the MDP describing the probabilistic evolution of the CPSp, where elementary predicates are expressed in terms of bigraphical configurations. In Fig. 6b, states where elementary predicates $Doctor.(Dev_{wlan}.(PNG))$ or $Nurse.(-_0)\big|Doctor.(-_1)$ are true are shown in dark gray. Essentially, Formula 6 expresses that either the nurse is co-located with the doctor within 50 time units with probability 0.8 in the physical space or a ping by the nurse successfully reaches the doctor's pager with probability 0.6. Reliability requirement R2 regarding the physical space as reflected in Formula 6 is violated in the configuration of Fig. 1. However, even minute changes in the design of the floor plan can have effects on requirement satisfaction in non-trivial ways. For example, merging patient rooms PT1 and PT4 would render property R2 satisfied.[4]

8 Run-time reasoning and adaptation

Satisfaction of certain requirements, such as requirements R3 and R4 in the hospital example, cannot be guaranteed at design time. Rather, satisfaction must be achieved at run-time by generating adaptive actions that can prevent the CPSp from violating requirements. Such an adaptive approach is based on (1) discovering possible requirements violations in future configurations of the CPSp that are determined by topological changes, and then (2) counteracting by applying actions that ensure requirements satisfaction.

Adaptation builds on a live representation of the topology of the CPSp characterizing a system operational environment modeled using BRS. Adaptation is then

---

[4] Example models can be found at home.deib.polimi.it/tsigkanos/smarthospital.

achieved by implementing the activities of the MAPE [37] (Monitoring, Analysis, Planning, Execution) loop, as shown in Fig. 2. Analysis and Planning are responsible for identifying possible requirements violations in future evolutions of the CPSp and generating an adaptation strategy, respectively. Monitoring and Execution are responsible for enacting the strategy at run-time. In the following, we illustrate how activities of the MAPE loop are configured for run-time adaptation of the hospital CPSp example of Fig. 1.

Monitoring During monitoring, events taking place in the CPSp corresponding to execution of actions by agents are received. For our hospital example, such events can indicate, e.g., access to a room by an agent or connections of mobile devices.

Analysis During analysis, future topological configurations of the space where requirements are violated are identified, if any. To support such kind of analysis on the evolution of the CPSp, the state space of possible reachable configurations is explored. As we discussed, the system state space is formally represented as an LTS, where states represent configurations and transitions represent actions occurring in the environment that generate new configurations.[5] Requirements R3 and R4 can then be expressed in branching time logic CTL [14] using bigraphical patterns as propositions expressing configurations:

$$\begin{aligned}\mathbf{R3}: &\ \mathtt{AG}\big(\neg Room_r.(Doctor.(-_0) \mid Patient \mid Visitor.(-_1) \mid -_2)\\ &\ \wedge Room_r.(Nurse \mid -_0)\big)\\ \mathbf{R4}: &\ \mathtt{AG}\big(Doctor.(Dev_{wlan,lnk}) \Rightarrow \neg(Visitor.(Dev_{wlan})\\ &\ \vee Doctor.(-_0) \mid Visitor.(-_1))\big).\end{aligned}$$

The state space can then be explored through model checking, looking for possible requirements violations. In Fig. 7, an LTS fragment shows how starting from a configuration corresponding to Fig. 1, the system evolves by executing actions that correspond to reactions modeled in the BRS. As illustrated in Fig. 7, starting from state *a*, the doctor may connect to patient data hosted on the server through the wireless network resulting to state *b*. Subsequently, if the visitor located in the waiting area enters PT4 and then PT5, or connects to the network, a violation of requirement R4 is triggered (states *c* or *d*, respectively). Additionally, if from the initial state the doctor enters with the patient room PT4 and then the waiting area when the visitor is there (state *e*), a violation of R3 occurs. In Fig. 7, states where requirements R3 or R4 are violated are shown in dark.

Planning If possible future requirements violations are detected, a counteracting adaptation strategy should be identified. Such a strategy is comprised of actions that the system managing the CPSp at run-time can take to prevent reaching states where

---

[5] To address scalability concerns, as an exhaustive generation and analysis of all LTS states may be impossible or inconvenient, analysis can be performed up to a lookahead horizon, corresponding to exploration of the execution of a number of actions by agents in the CPSp. If that horizon is reached, analysis is then performed again.

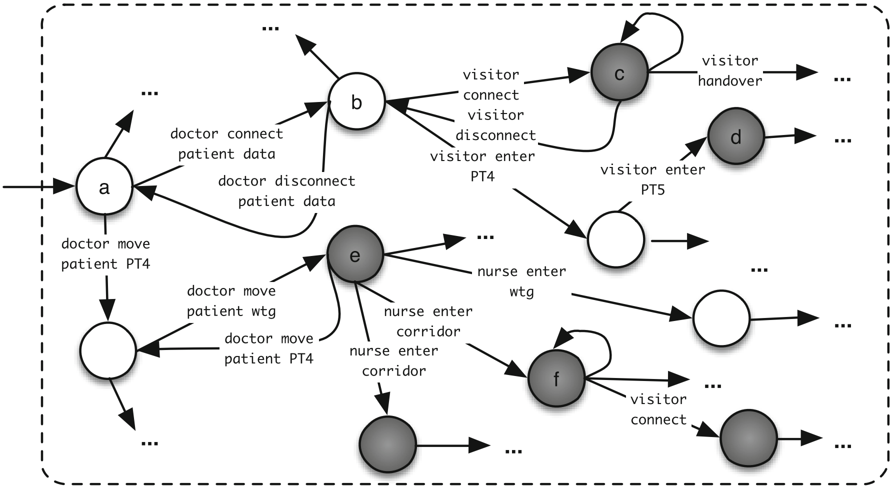

Fig. 7 Fragment of an LTS used for analysis and planning at run-time. States represent configurations of the CPSp, and transitions show how configurations change due to reactions modeled in the BRS

configurations are violating. For example, for state *c*, the controller can preemptively forbid connections to that wireless network by visitors before reaching that state, or in case this is not possible, force disconnection or handover of the visitor connection to another wireless network, if state *c* is reached. Similarly, for violating state *e*, the system can prompt the nurse to enter the waiting area so that she is co-located with the visitor, or notify the doctor to move to another room. The choice among different possible counteractions can be enabled by a suitable classification of available actions to the planning activity.

Execution Execution amounts to enacting the counteractions produced by planning. The reader may refer to [60] for an example of how execution can be implemented in practice.

9 Related work

Research in diverse areas has considered analysis of building designs with respect to requirements. In this section, we discuss key approaches. First, we consider rule-based checking of building designs in the AEC industry as well as approaches supporting the analysis of specific dynamic scenarios. Subsequently, we discuss a similar line of research in that it aims at an extension of traditional BIM towards modeling and analysis of smart built environments. Thereafter, we consider spatial assistance systems as another related line of research which partially addresses also building designs. Finally, we review related work which shares similarities with our approach for at least one major aspect; this includes the use of graphs and bigraphs as formal building models. The objective is not to be exhaustive, but to position our work on cyber-physical spaces with respect to relevant lines of research followed by larger research communities.

Rule-based checking of building designs Rule-based systems [21,55] in the AEC domain assess building designs according to various criteria, expressed as rules, constraints or conditions. IFC-based BIM frameworks such as Solibri or Tecla platforms [54,57] utilize rule-based checking, including a variety of functions, providing capabilities for checking a model from simple checks such as shape overlappings, existence of specific objects as well as more detailed checks based on ISO accessibility regulations or fire code path distances. Additionally, more advanced techniques of automated rule-based safety checking [68] have been implemented over APIs of such frameworks. BERA is such a DSL[41] where one can efficiently define and check rules on a building design, with an implementation portable to different BIM platforms. Our approach differs from current rule-based approaches in two directions. First, it views rule-based checks in the general context of formal (static and dynamic) semantics of the modeling formalism. Second, it also accounts for CPSp dynamics, which includes cyber aspects critical to complex requirements that span both spaces.

Analysis of dynamic scenarios Model-based reasoning techniques have been utilized for studying and simulating certain dynamic aspects of building information models. However, existing approaches are ad hoc and focus on very specific scenarios, such as building evacuation [24,27] or fire response management processes [31]. The proposed analyses cannot be adopted to generally reason about dynamic properties of a CPSp as it is possible using our approach. A more generic approach has been presented by Isikdag et al. [32]. It describes a BIM-oriented modeling methodology which extends standard BIM/IFC by providing detailed semantic information intended to support the analysis of general indoor navigation requirements. The approach allows designers of spatial environments to formulate SQL-like queries to extract semantic information considered relevant w.r.t. an indoor navigation requirement. In contrast to our approach to automatically checking dynamic properties of a space against some formally defined requirements, however, this can only be used as a semi-automatic utility.

Extending BIM to smart built environments Concerning the conceptual modeling of smart built environments, an approach for extending BIM by smart objects has been proposed in [67]. The modeling approach is conceptually at a lower level and is largely complementary to ours. It focuses on the modeling of different kinds of sensors and actuators and their integration into the physical space, while we are mostly interested in high-level conceptual entities along with their topological relationships and connections in the physical space. The approach follows a tight integration of the BIM extensions into standard BIM modeling tools and presents a prototypical implementation of a tool chain based on the Revit CAD tool. Moreover, some forms of both design- and run-time analyses have been outlined based on the cyber-extended BIM. However, similar to the traditional analysis tools, they target specific scenarios instead of providing a generally applicable methodology. At design time, the allocation of sensors and actuators within the physical layout can be analyzed and potentially optimized. Dynamic analysis and adaptation at run-time is supported by a facility management tool which aims at optimizing energy management of a building being connected to a smart grid.

Spatial assistance systems Spatial assistance systems in general and architectural design assistance systems in particular have been developed, e.g., in [6]. Most notably w.r.t. our approach, a formal modeling approach for architectural design has been proposed recently [9]. Based on an ontological model of architectural domain knowledge, architects may specify instances, i.e., concrete building designs, and employ reasoning services provided by the assistance system [52]. The project focuses on people-centered architectural design qualities which can be characterized as visuo-spatial and navigational experience of building users, subjective lighting influences as well as navigation and orientation patterns being typical examples of this. These qualities are orthogonal to the regulatory requirements and other safety, reliability and security requirements for which we aim to provide formal assurances. Moreover, the run-time dimension is not considered by the approach.

Graphs as formal building models and case-based design Different forms of graphs as formal models of static representations of buildings have been proposed by several approaches in diverse fields such as architecture informatics [38] or computer graphics [66], with different objectives. Most of the approaches target case-based reasoning [1] in the architectural domain. In [38], for instance, the topology of spatial configurations is extracted from building information models as well as handwritten architectural sketches [2] and represented as graphs. Focusing on security reasoning while aiming at early design phases, Porter et al. [47] propose a method and heuristics to discover security threats on building specifications via simulation, utilizing BIM. Analyses such as similarity checking are performed based on graph matching techniques [15]. The overall goal is to build a comprehensive case base which can be queried to retrieve previously designed and stored building designs serving as reference examples in early design stages. Similar case-based reasoning techniques using different information and analyses have been proposed in the literature; surveys can be found in [30,50]. Case-based design is largely complementary to our approach. In fact, static and dynamic models used within our approach, notably bigraphs and bigraphical reactive systems, can be considered as another source of information which may be integrated and analyzed in existing case bases.

Adoption of bigraphs and BRS This paper builds on previous work that has used BRS to provide formal semantics for Building Information Models [58]. The adoption of BRS for modeling physical spaces has been considered elsewhere in the literature. Walton et al. [64] focus on BRS as a formal modeling approach to represent indoor spaces and mobility of objects and agents in those spaces; this work aims at reasoning about path-based navigation tasks, i.e., reachability of specific locations by agents. BiAgents [46] are a formalism utilizing bigraphs for modeling the physical space and abstract algebraic structures for the cyber space and have been used to identify strategies to prevent ill-defined concurrency situations that can emerge from cyber agents operating in shared physical structures. Additionally, a form of BRS has been used [12] to model and reason about structure and connectivity in network topologies and management systems.

10 Conclusion and outlook

Nowadays spatial environments are dynamic cyber-physical spaces in which the traditional physical world and the digital world are heavily interwined and interacting with each other. This leads to new requirements, e.g., concerning security, reliability, safety etc., which have to be considered during design and operation of a CPSp, a great concern in mission-critical spaces such as smart hospitals or industrial plants. Apart from rule-based checking of static properties of physical space plans typically created using CAD software environments, the current practice in the construction industry is weak in mitigating the challenges arising from cyber-physical spaces in a holistic manner. This applies in particular to the challenges arising from the general dynamism that cyber-physical spaces exhibit an aspect which has been of minor importance in traditional physical spaces, except for some exceptional cases such as building evaluation scenarios, simulated using dedicated software solutions.

In this paper, we presented our vision of how to support designers in the challenging task of designing and managing operation of dynamic cyber-physical spaces adopting software engineering principles. To put it bluntly, this can be considered as a second happy marriage of two disciplines which, although apparently being far apart from each other, share a considerable amount of challenges and body of knowledge. Formerly, software engineering as the considerably younger discipline has learned a lot from traditional architecture, e.g., the very notion of an “architecture” or the usage of “architectural patterns” as proven solutions for recurring kinds of problems. Conversely, in this paper we argue and show how the architecture discipline can profit from recent advances in software engineering. In short, we argue that methods from the fields of formal verification and self-adaptive systems can play an important role in the design and operation of cyber-physical spaces.

Early results from experiments conducted based on prototypical implementations are promising and demonstrate the potential benefits of our approach. However, much work has to be done to fully realize our vision of a holistic approach to modeling, analysis, and operation of cyber-physical spaces. For example, many steps of our overall process resort to manual interventions, and an integration of our prototypical solutions into state of the art tool chains as used in the construction industry is of primary importance. Such an integration provides the setting in which we have to evaluate our approach using real-world case studies and involving domain experts from the architectural domain.

Furthermore, we still have a long way to go in providing full support to design-time and run-time analyses. At design time, the modeling tooling should fully support exploratory design, where trade-offs between different design alternatives can be analyzed and designers can be given some form of guidance in selecting the proper ones. Model checking at design time can indicate that certain topological configurations may be entered which violate requirements. This is highly useful, but it does not provide guidance to selecting alternative design choices that would remedy the problem. It may also be useful to provide a repertoire of action primitives that the designer may perform to alter a configuration, such as adding a door, placing or moving entities. By modeling design-time operations, state exploration can be utilized for the purpose of finding sequences of such edit operations that can modify a model under construction as to exhibit no requirement violations.

Considerable research is also needed to fully support run-time verification and adaptation. For example, full support to modeling and reasoning about timing issues is still lacking in the approach we presented in this paper. For example, whenever at run-time a possible requirements violation is detected, in the adaptation phase we do not consider how the time profile of actions occurring at run-time, which lead to topological changes, can affect the selection of possible countermeasures, which are characterized by their own inertia.

In conclusion, we believe that software engineering has a lot to contribute to making design and operation of CPSp successful and dependable. A long and challenging research avenue, however, is still ahead of us.

Acknowledgments We wish to thank Bashar Nuseibeh and Liliana Pasquale for inspiring contributions to many points touched in this paper and for continuous cooperation. Claudio Menghi also supported some initial phases of this work. Finally, we wish to thank Ezio Arlati for helping us understand the world of architecture and his enthusiasm in collaborating to form an interdisciplinary workgroup at Politecnico di Milano, joining efforts from Architecture and Computer Science. This work was partially supported by ERC Advanced Grant No. 227977 (SMScom).

References

1. Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun 7(1):39–59
2. Ahmed S, Weber M, Liwicki M, Langenhan C, Dengel A, Petzold F (2014) Automatic analysis and sketch-based retrieval of architectural floor plans. Pattern Recogn Lett 35:91–100
3. Alshanqiti A, Heckel R (2015)Extracting visual contracts from java programs. In: Proc. 30th Intl. Conf. on Automated Software Engineering. IEEE, pp 104–114 (2015)
4. Baheti R, Gill H (2011) Cyber-physical systems. Impact Control Technol 12:161–166
5. Baier C, Katoen JP et al (2008) Principles of model checking, vol 26202649. MIT press, Cambridge (2008)
6. Barkowsky T, Bateman JA, Freksa C, Burgard W, Knauff M (2005) Transregional collaborative research center SFB/TR 8 spatial cognition: Reasoning, action, interaction. IT 47(3):163–171
7. Bass L, Weber I, Zhu L (2015) DevOps: A Software Architect's Perspective. Addison-Wesley Professional, USA
8. Bennett KH, Rajlich VT (2000) Software maintenance and evolution: a roadmap. In: Proceedings of the Conference on the Future of Software Engineering. ACM, New York
9. Bhatt M, Hois J, Kutz O (2012) Ontological modelling of form and function for architectural design. Appl Ontol 7(3):233–267
10. Birkedal L, Damgaard TC, Glenstrup AJ, Milner R (2007) Matching of Bigraphs. Electron Notes Theor Comput Sci 175(4):3–19
11. Brambilla M, Cabot J, Wimmer M (2012) Model-Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, USA
12. Calder M, Koliousis A, Sevegnani M, Sventek J (2014) Real-time verification of wireless home networks using bigraphs with sharing. Science of Computer Programming 80:288–310
13. Chen K, Sztipanovits J, Abdelwalhed S, Jackson E (2005) Semantic anchoring with model transformations. In: Model Driven Architecture–Foundations and Applications. Springer, New York, pp 115–129 (2005)
14. Clarke EM, Grumberg O, Peled DA (1999) Model checking. MIT press, Cambridge
15. Conte D, Foggia P, Sansone C, Vento M (2004) Thirty years of graph matching in pattern recognition. Int J Pattern Recogn Artif Intell 18(03):265–298
16. Czarnecki K, Helsen S (2006) Feature-based survey of model transformation approaches. IBM Syst J 45(3):621–645

17. Day M (2008) The move to bim with archicad, vol 12. AEC Magazine, London
18. De Lemos R, Giese H, Müller HA, Shaw M, Andersson J, Litoiu M, Schmerl B, Tamura G, Villegas NM, Vogel T et al (2013) Software engineering for self-adaptive systems: a second research roadmap. Springer, New York
19. Depoy J, Phelan J, Sholander P, Smith B, Varnado G, Wyss G (2005) Risk assessment for physical and cyber attacks on critical infrastructures. In: Military Communications Conference. MILCOM. IEEE, pp 1961–1969
20. Eastman C, Eastman CM, Teicholz P, Sacks R (2011) BIM Handbook: A Guide to Building Information Modeling for Owners, Managers ,Designers, Engineers and Contractors. J.W & S
21. Eastman C, Lee JM, Jeong YS, Lee JK (2009) Automatic rule-based checking of building designs. Autom Constr 18(8):1011–1033
22. Finkelstein A, Kramer J, Nuseibeh B, Finkelstein L, Goedicke M (1992) Viewpoints: A framework for integrating multiple perspectives in system development. Int J Softw Eng Knowl Eng 2(01):31–57
23. Forejt V, Kwiatkowska M, Norman G, Parker D (2011) Automated verification techniques for probabilistic systems. In: Formal Methods for Eternal Networked Software Systems. Springer, New York, pp 53–113
24. Gianni D, Bocciarelli P, D'Ambrogio A, Iazeolla G (2015) A model-driven and simulation-based method to analyze building evacuation plans. In: Proceedings of the 2015 Winter Simulation Conference. IEEE Press, New York, pp 2644–2655
25. Godfrey MW, German DM (2008) The past, present, and future of software evolution. In: Frontiers of Software Maintenance, 2008. FoSM. IEEE, pp 129–138
26. Group OM (2013) Meta object facility (mof) core specification, version 2.4.1. OMG document number: formal/2013-06-01
27. Hamacher HW, Tjandra SA (2001) Mathematical modelling of evacuation problems: A state of art. Fraunhofer-Institut für Techno-und Wirtschaftsmathematik (ITWM), Fraunhofer
28. Hansson H, Jonsson B (1994) A logic for reasoning about time and reliability. Formal Aspects Comput 6(5):512–535
29. Hermanns H (2002) Interactive markov chains. The Quest for Quantified Quality, vol 2428. Springer, Berlin, Heidelberg
30. Heylighen A, Neuckermans H (2001) A case base of case-based design tools for architecture. Comput Aided Des 33(14):1111–1122
31. Isikdag U, Underwood J, Aouad G (2008) An investigation into the applicability of building information models in geospatial environment in support of site selection and fire response management processes. Adv Eng Inform 22(4):504–519
32. Isikdag U, Zlatanova S, Underwood J (2013) A BIM-oriented model for supporting indoor navigation requirements. Comput Environ Urban Syst 41:112–123
33. ISO 16739: Industry Foundation Classes (IFC) (2013) Data Sharing in the Construction and Facility Management Industries. http://iso.org/iso/home/store/catalogue_tc/catalogue_detail.html?csnumber=51622
34. Kappel G, Langer P, Retschitzegger W, Schwinger W, Wimmer M (2012) Model transformation by-example: a survey of the first wave. In: Conceptual Modelling and Its Theoretical Foundations. Springer, New York, pp 197–215
35. Kehrer T, Taentzer G, Rindt M, Kelter U (2016) Automatically deriving the specification of model editing operations from meta-models. In: Proc. Intl. Conf. on Model Transformations (2016) (to appear)
36. Kehrer T, Tsigkanos C (2016) An emof-compliant abstract syntax for bigraphs. In: Graphs as Models at ETAPS'16 (2016) (to appear)
37. Kephart JO, Chess DM (2003) The vision of autonomic computing. Computer 36(1):41–50
38. Langenhan C, Weber M, Liwicki M, Petzold F, Dengel A (2013) Graph-based retrieval of building information models for supporting the early design stages. Adv Eng Inform 27(4):413–426
39. Larrea-tamayo H (2016) Cambridge, M.U.L.L.Y.W.M.U.L.K.B.M.U.L.J.W.B.C.U.: Apparatuses, systems, and methods for transformable living spaces (2016). http://www.freepatentsonline.com/y2016/0031090.html
40. Lee EA (2008) Cyber physical systems: Design challenges. In: Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on. IEEE, pp 363–369
41. Lee JK, Eastman CM, Lee YC (2015) Implementation of a bim domain-specific language for the building environment rule and analysis. J Intell Robot Syst 79(3–4):507–522
42. Lehman MM (1980) Programs, life cycles, and laws of software evolution. Proc IEEE 68(9):1060–1076

43. For Medicare & Medicaid Services C et al (1996) The health insurance portability and accountability act of 1996 (hipaa). http://www.cms.hhs.gov/hipaa
44. Milner R (2009) The Space and Motion of Communicating Agents. Cambridge University Press, Cambridge
45. Nakashima H, Aghajan H, Augusto JC (2009) Handbook of ambient intelligence and smart environments. Springer Science & Buss. Media, New York
46. Pereira E, Kirsch C, Sengupta R (2012) BiAgents—A Bigraphical Agent Model for Structure-aware Computation. Cyber-Physical Cloud Computing Working Papers. CPCC Berkeley, USA
47. Porter S, Tan T, Tan T, West G (2014) Breaking into bim: Performing static and dynamic security analysis with the aid of bim. Autom Constr 40:84–95
48. Rajkumar RR, Lee I, Sha L, Stankovic J (2010) Cyber-physical systems: the next computing revolution. In: Proceedings of the 47th Design Automation Conference. ACM, USA, pp 731–736
49. Rappaport TS et al (1996) Wireless communications: principles and practice, vol. 2. Prentice Hall PTR, New Jersey
50. Richter K, Heylighen A, Donath D (2007) Looking back to the future-an updated case base of case-based design tools for architecture. Knowl Modell eCAADe 25:285–292
51. Salehie M, Tahvildari L (2009) Self-adaptive software: Landscape and research challenges. ACM Trans Auton Adapt Syst 4(2):14
52. Schultz CPL, Bhatt M (2010) A multi-modal data access framework for spatial assistance systems: use-cases with the building information model (BIM/IFC). In: Indoor Spatial Awareness - ISA 2010, 2nd International Workshop. Proceedings. ACM, San Jose, pp 39–46
53. Sha L, Gopalakrishnan S, Liu X, Wang Q (2008) Cyber-physical systems: A new frontier. In: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2008), 11-13 June 2008, Taichung. IEEE Computer Society, Taiwan, pp 1–9
54. Solibri (2016) Solibri Model Checker. http://www.solibri.com
55. Solihin W, Eastman C (2015) Classification of rules for automated bim rule checking development. Autom Constr 53:69–82
56. Stahl T, Völter M, Bettin J, Haase A, Helsen S (2006) Model-driven software development - technology, engineering, management. Pitman. http://dblp.uni-trier.de/rec/bib/books/daglib/0016398
57. Tekla (2016) Tekla Structures. http://www.tekla.com/products/tekla-structures
58. Tsigkanos C, Kehrer T, Ghezzi C, Pasquale L, Nuseibeh B (2016) Adding static and dynamic semantics to building information models. In: Proceedings of the 2nd international workshop on software engineering for smart cyber-physical systems. ACM, pp 1–7
59. Tsigkanos C, Pasquale L, Ghezzi C, Nuseibeh B (2015) Ariadne: topology aware adaptive security for cyber-physical systems. In: Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on. vol 2. IEEE, pp 729–732
60. Tsigkanos C, Pasquale L, Ghezzi C, Nuseibeh B (2015) Ariadne: Topology Aware Adaptive Security for Cyber-Physical Systems. In: Proc. of the 37th International Conference on Software Engineering, pp 729–732
61. US Department of Veterans Affairs, Veterans Health Administration (2016) PG-18-9 Space Planning Criteria, Medical—Surgical Inpatient Units. http://cfm.va.gov/til/space/SPchapter100.pdf
62. Van Deursen A, Klint P, Visser J (2000) Domain-specific languages: an annotated bibliography. Sigplan Notices 35(6):26–36
63. Vogel-Heuser B, Feldmann S, Folmer J, Kowal M, Schaefer I, Ladiges J, Fay A, Haubeck C, Lamersdorf W, Lity S et al (2015) Selected challenges of software evolution for automated production systems. In: Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on. IEEE, pp 314–321
64. Walton LA, Worboys M (2012) A qualitative bigraph model for indoor space. In: Geographic Information Science. Springer, New York
65. Weiser M (1991) The computer for the 21st century. Sci Am 265(3):94–104
66. Wessel R, Blümel I, Klein R (2008) The room connectivity graph: Shape retrieval in the architectural domain. In: The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2008. UNION Agency-Science Press, USA
67. Zhang J, Seet BC, Lie TT (2015) Building information modelling for smart built environments. Buildings 5(1):100–115
68. Zhang S, Teizer J, Lee JK, Eastman CM, Venugopal M (2013) Building information modeling (bim) and safety: Automatic safety checking of construction models and schedules. Autom Constr 29:183–195