Identifying Visualization Opportunities to Help Architects Manage the Complexity of Building Codes

Stan Nowak, *Simon Fraser University, Surrey, BC, CAN*
Bon Adriel Aseniero, *Autodesk, Toronto, ON, CAN*
Lyn Bartram, *Simon Fraser University, Surrey, BC, CAN*
Tovi Grossman, *University of Toronto, Toronto, ON, CAN*
George Fitzmaurice, *Autodesk, Toronto, ON, CAN*
Justin Matejka, *Autodesk, Toronto, ON, CAN*

Abstract—
We report a study investigating the viability of using interactive visualizations to aid architectural design with building codes. While visualizations have been used to support general architectural design exploration, existing computational solutions treat building codes as separate from, rather than part of, the design process, creating challenges for architects. Through a series of participatory design studies with professional architects, we found that interactive visualizations have promising potential to aid design exploration and sensemaking in early stages of architectural design by providing feedback about potential allowances and consequences of design decisions. However, implementing a visualization system necessitates addressing the complexity and ambiguity inherent in building codes. To tackle these challenges, we propose various user-driven knowledge management mechanisms for integrating, negotiating, interpreting, and documenting building code rules.

Building codes form a critical part of architectural design practices, defining the external constraints architects must work within to ensure their designs meet standards for safety, accessibility, and sustainability among other factors. With the advent of Computer-aided Design (CAD) and Building Information Modeling (BIM) tools, automating building code compliance checks has received considerable attention [1]. However, automation initiatives neglect the needs of architectural designers who use building codes not only for compliance checking but as input for design [2]. While interactive visualizations have been used to support design space exploration in architectural parametric design applications [3], little visualization work has focused specifically on building codes. Our research gives a preliminary understanding of how visualization could better support the needs of architects that design with building codes in mind, what challenges they face, and how they might be addressed.

XXXX-XXX © 2023 IEEE
Digital Object Identifier 10.1109/XXX.0000.0000000

We conducted a series of qualitative participatory interview, survey, and design probe studies to identify tractable visualization challenges for architects' work practices around building codes. Our expert participants thought visualizations would be most appropriate for helping them maintain awareness of code constraints at the early stages of design involving many spatial and quantitative dependencies that can be challenging to track. By revealing the design options available within building code constraints and providing feedback on the potential consequences of future actions, visualizations can support design planning, execution, and evaluation. However, building codes are constantly rewritten, vary by jurisdiction, result in conflicts, and vary in interpretation. This complexity makes computational pre-specification of code logic problematic. Instead, we argue architects themselves should configure code logic, derived measures driving visualizations, and reference materials aiding code interpretation. We explore a variety of user-driven knowledge management mechanisms to address these challenges.

While our work highlights opportunities for visualization in this domain, it also shows how the success or failure of any such systems depends on how well they integrate into broader work practices and contexts beyond what would be conventionally considered relevant for the design of visualization tools. There is a latent need for visualization solutions to deal with ambiguous and complex data, to enable direct access to underlying data, and to accommodate both external and user-generated contextual knowledge.

We contribute:

- Insights from design studies with architects characterizing visualization needs, opportunities, and challenges to support architectural design using building codes.
- Design implications for visualizations to support design exploration and user-driven knowledge management.

BACKGROUND

Building Codes

Building codes are laws that regulate the safe construction of buildings, often taking the form of rule-based constraints for spatial dimensions, geometry, or construction materials among other factors [4]. While the specifics of building codes vary by jurisdiction - such as cities or countries - their logic and procedures are largely similar. Architects submit building plans to regulatory government authorities. If the plans comply with building codes, the authorities issue building permits, which legally authorize construction. Frequent updates to codes and overlapping jurisdictions with different codes can result in conflicting requirements that architects must interpret and resolve.

The interpretation and application of building codes is usually centred around marked-up architectural drawings. Architects, compliance officers, and other professionals, including consultants specializing in building code interpretation, exchange such drawings to aid discussions on the application and interpretation of building codes. Markup might include specific measurements used to demonstrate code compliance or comments from various stakeholders.

Neglecting the Needs of Architects

Given the arduous compliance checking process, automated compliance checking software has been explored for decades but with little to no material outcomes [5], [6]. The primary impediment is that building codes are often ambiguous and are deliberately written in a way that empowers authorities to apply their judgment to any given case. For instance, consider the following building code: *"...The door shall be openable from both sides without special knowledge and effort..."* [7] Interpreting what *"special knowledge and effort"* means requires consideration of context-dependant factors of individual cases. Even in jurisdictions that have rewritten laws to enable automated compliance checking [1], practical application is challenged by the volume of additional BIM meta-data that must be entered and the "black box" nature of algorithms. These factors can make architects distrustful of such systems and dissuade them from use.

Such automation efforts have also neglected the design practices of architects. Automation initiatives often mistake design and compliance as separate steps when they are closely related. Recent work has explored practical approaches for user-driven governance of automated compliance checking during rather than only after design [8]. However, architects' needs extend beyond compliance checking. A study investigating architects' perspectives on automation for code compliance in healthcare building projects found that architects find value in automation only for simple repetitive tasks [2]. They suggest that supportive technology should provide relevant information to aid design decision-making and support the rapid exploration of design alternatives. Visual analytics is well-suited to provide this task. Researchers have used visualization to aid design space exploration in other parametric design applications [3].

Design Sensemaking

Design in architecture is described as *"a goal-oriented, constrained, decision-making, exploration, and learning activity ... that depends on the designer's perception of the context"* [9]. Designers explore a "design space" of potential solutions often relying on external representations to aid navigation and allow inferences about potential solutions without the need to actively create them [10]. Often this takes the form of generating, comparing, and evaluating alternatives according to specific constraints [3].

Visualization in Architecture. Visualization plays a central role in CAD and BIM tools [11]. Visualization research has explored a variety of ways to support design exploration in parametric [12], and generative design applications [3]. Despite architects' expressed desire for design space exploration support [2], visualization of building codes has primarily focused on highlighting non-compliant building elements using automated rule-checking [11], [17]. Our research aims to fill this gap by exploring opportunities for visualizations to better support architects' needs as they relate to designing with building codes.

STUDY

Following standard methods used to characterize problems for visualization [13], we engaged with experienced architects to explore issues and potential solutions. We conducted semi-structured interviews to solicit their perspectives and ideas on visualization application areas, potential challenges, and existing approaches. Subsequently, we conducted a survey involving a broader sample of experts to refine and check our understanding. Drawing on the resulting findings, we then created a set of low-fidelity mock-ups as design probes to explore the viability of potential solutions.

Participants

Using a snowball sampling approach, we recruited participants with a background in architecture, some of whom hold positions at a large architectural and construction software company. We chose participants with this background to provide insightful input on technology design.

Fifteen architects (3 female and 12 male) ranging by career stage, architectural firm size, and countries worked in, participated. The continuity of existing participants helped us check our understanding throughout all studies, while a stream of new participants helped solicit new perspectives and refine our findings (**Table 1**).

INTERVIEW & SURVEY

Procedure

One researcher conducted semi-structured interviews lasting 40-60 minutes over Zoom[1]. They asked questions about the biggest challenges architects face related to building codes, the role building codes play in design, the challenges of implementing interactive technologies to deal with building codes, and what an ideal system to visualize building codes in a CAD program might look like. Data collected include notes taken during interviews, interview video recordings, and transcripts from audio recordings.

We distributed surveys via Typeform[2] in which we asked participants to respond to a series of statements about their level of agreement via a 5-point Likert scale. Following such questions, we invited participants to explain their answers in an open-ended format. In addition, we asked several other open-ended questions.

Analysis

The research team met regularly to discuss quotes and findings, developing four themes that one researcher used to code transcripts. A second researcher coded a 10% sample of transcripts. Conflicts were discussed and resolved by updating coding guidance and existing codes.

We treat survey findings qualitatively, interpreting relative agreement using clarifying open-ended responses. Rather than generalizing results to the broader population of architects, we use numbers as a weak indicator and focus primarily on the diverse perspectives that our questions elicited to validate our own understanding.

Findings

We summarize findings from both interviews and surveys below. We organize sections according to broad topics of inquiry and bold core themes identified in analyses.

Where can visualization help?

In both interviews and the survey, participants suggested the main challenge visualizations could improve is in **maintaining awareness** of building codes during the early stages of design. These include heavily spatial reasoning, involving planning the overall structure of buildings, arranging the layout of rooms, and iteratively creating and evaluating alternative designs. Evaluation often involves architects repeatedly manually recalculating building code constraints to ensure compliance and avoid costly redesigns. *"It's a constant struggle to recalculate . . . if you aren't particularly rigorous . . . it could surprise you later." (P3).* They suggested a tool providing feedback about compliance could reduce errors. *"If a tool helps them knock off*

[1] https://zoom.us/

[2] https://www.typeform.com/

TABLE 1. Table of participants

ID	Age	Background	Architectural Org. Size	Countries with Code Experience	Interview	Survey	Design Probe
P1	38	Architect, Program Manager, Educator	6-80	USA, Canada, Germany, Italy	x		
P2	49	Architect, Product Manager	40-1000+	USA	x	x	x
P3	30	Architect, Product Owner	50	USA	x	x	x
P4	42	Architect, Marketing Manager	20+	USA	x		
P5	33	Architect	6-150	Canada	x	x	x
P6	33	Architect, Technical Solutions Architect	35-900	USA, Turkey	x	x	x
P7	39	Architect, Technology Consultant	15-300	USA	x	x	x
P8	42	Architect, Research Scientist	40-1000+	USA, Canada	x	x	x
P9	35	Architectural Project Manager, Sales Manager	40-1000+	USA	x	x	
P10	64	Architect, Principal QA Analyst	15-200+	USA		x	x
P11	N/A	Architect	270	Canada		x	
P12	30	Architect	15	Canada		x	x
P13	29	Architect	40	Canada		x	x
P14	49	Architect, Technical Marketing Manager	1000+	USA, UK, El Salvador		x	x
P15	33	Architect	20	Canada		x	x

some of the most obvious things in advance, it means real dollar savings down the line." (P1).

We refined our understanding through surveys (**Fig. 1**). Nearly all architects agreed that real-time feedback about building codes would be useful and only P15 was neutral (Fig 1.A). P15 clarified that visualizing life safety codes would be useful in early design stages. By contrast, some agreed or expressed neutrality that the existing segmented workflows were efficient (Fig. 1.B). P12, P11, and P5 explained that for some more complex building codes, this segmented workflow is necessary. P12 said it allows an *"iterative process"* with *"options"* for code compliance where automated software-based checking may be too *"rigid"* and may not permit alternative interpretations of how compliance may be met. We were surprised to find that many architects did **not** agree it was easy to lose track of code considerations (Fig 1.C). Participants explained this depends on skill (P9, P14), that codes are already key considerations (P12, P8, P15), and that any mistakes would be caught at later stages (P11). Considering these participants' other survey responses, which express a generally positive outlook on real-time feedback and the potential for visualizations to *"streamline development"* (P15), it appears that they believe that entirely forgetting about certain building codes might only be a challenge for some junior architects, with mistakes being caught later in the process. Taken together, architects perceive potential benefits from visualization in a limited set of circumstances but also articulate challenges and concerns threatening the viability of visualizations as a solution. We expand on these issues in the sections below.

How can visualization help?

Architects articulated two high-level visualization strategies to help them maintain awareness of building codes during design. First, they described the use of visual markup to highlight compliance issues and drew analogies to linting used for spell-checking in word processor systems. *"I almost see it like it's a spell checker." (P6).* Secondly, they wanted to see the amount of space they have available between the dimensions of their design and the constraints set by building codes in order to make informed design decisions that align with goals, such as maximizing the floor space for rental purposes. *"How much to be rentable. . . to meet a return on investment. . . how much corridor space do we have to build for, because that's not rentable space." (P9).* They provided other examples of constraints, such as general floorplan requirements set by clients, that have to be counterbalanced against building codes. *"[Program areas] usually come in the form of a document that lists all the spaces and the sizes of the spaces that a district would like to have . . . it's difficult to count up [rooms and spaces within the design] and compare that to the document you are provided." (P3).* While highlighting compliance issues in CAD is not new, we are not aware of any previous efforts to visualize the state of a design relative to constraints defined by building codes in this way.

We presented both these visualization strategies to a broad sample of architects in surveys (Fig. 1.D-E). Feedback was generally favourable. One architect thought showing "available space" would aid *"anticipating the impact of code in design solutions [and] facilitate design decision making and design review." (P14).* P8 thought that just like other *"spell-check"* tools, there would be value, but *"grammar and higher ideas"* would be missed, highlighting the challenges of translating building codes to quantitative data.

What concerns do architects have about visualization?

Drawing on prior experiences with technology-driven solutions, architects expressed concern that visualization systems would try to automate and replace their practices, potentially undermining their sense of **trust, control, and creativity**. They frequently expressed concern that computational or automated solutions

This article has been accepted for publication in IEEE Computer Graphics and Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MCG.2023.3307971

may impede their control over the creative design process or cause issues of trust due to a lack of transparency in how a system is interpreting and applying codes. Citing experience with some parametric design software, architects thought that automated solutions might restrict their actions or act on their behalf. P1 thought that over-emphasized visual feedback may lead to fixation and reduce creativity. *"What is the right balance that's not going to impede creativity"*. P2, a former architect with experience in projects using automation in architectural design shared insights about the importance of making data processing transparent to ensure trust. *"Architects . . . want to understand how it arrived at a calculation so they can trust it"*.

Architects were nearly unanimous in expressing their need for **agency** (Fig. 1.F), especially as it pertains to exploring alternative solutions. *"I think visual cues would be very beneficial but not allowing 'non-code compliant' choices would be a frustrating experience. [There's] many different ways to achieve compliance and the interpretation the program takes could be incorrect for certain scenarios." (P12).*

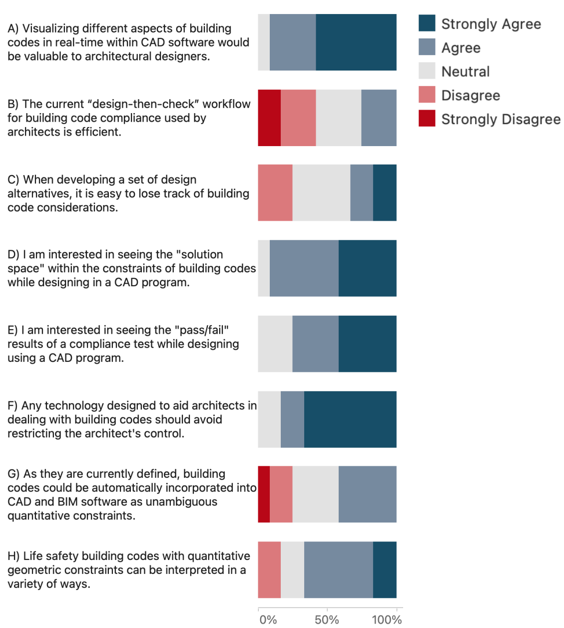

FIGURE 1. Survey question exploring issues related to building code visualization.

What challenges stand in the way?

The key challenge for a building code visualization system to address is the **ambiguity and complexity** of building codes. This poses a major impediment to purely computational or automated solutions.

Interpretations of how to translate legal language into specific design constraints vary and can change throughout a project. *"Even at the code authority level, it's the person who is interpreting . . . for building permits . . . the next building code official will . . . change their minds." (P7).* In surveys, most participants agreed that even quantitative geometric constraints can be ambiguous (Fig. 1.H) and disagreed that current building codes could be automated and unambiguous (Fig. 1.G). P14 explained that this ambiguity is necessary as many projects present novel challenges, such as safety considerations for new technologies, that would be difficult to anticipate. "*Codes may be perceived as a restraint to design exploration and imagination. Codes must allow for an option to be scientifically challenged, allowing new technology to be properly incorporated."* Meanwhile, P2 argued that a *" 'standard interpretation' could be agreed upon by at least 85% of cases."*

Building code laws themselves are frequently updated. *"Building codes change so often that what worked five years ago . . . is no longer viable across the street." (P7).* One architect discussed their experience developing a script to automate compliance checking. The script quickly became outdated, highlighting the need for customization in automated solutions. This also underscores the challenges of maintaining such automated compliance systems.*"Things don't scale well in this space unless you expose the capability to write your own version of that code. [our project] was valid for one form [and] for maybe a year . . . you would have to . . . rewrite the entire piece." (P7).*

Architects must resolve conflicts resulting from interactions between building codes or unique circumstances allowing exceptions to be made. *"[Exceptions are] really key to a lot of this stuff . . . there's so many things that you can't do if you just read the building code as is." (P7).* Such conflicts must be negotiated with authorities. *"It becomes a very subjective game that you then have to support with evidence." (P9).* P11 noted that any software would need to be able to allow the architect to *"triage requirements"*

This complexity demands a variety of **knowledge management** strategies to gather, discuss, document and disambiguate building code requirements. Architects discussed how time-consuming navigating code requirements can be as well as the procedures they use for easier information retrieval. *"People would keep Excel spreadsheets [with references to the relevant] applicable code [others] would just tag it in the actual book." (P9).* The exchange of marked-up drawings, often in PDF form, can be difficult to navigate but facilitates knowledge transfer about building codes and solutions between senior and junior architects. *"My*

This article has been accepted for publication in IEEE Computer Graphics and Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MCG.2023.3307971

supervisor would go in and redline everything . . . then I go back into CAD and check off each red line as I go." (P5).

DESIGN PROBES

Based on findings from the above studies, we created low-fidelity design probes [14] to explore potential visualization design solutions with architects. We explore: 1.) How visualizations can provide feedback about building codes using both the strategies of showing "available space" within design constraints and highlighting compliance issues; and 2.) How a variety of user-driven knowledge management strategies could address issues of ambiguity and complexity that challenge the technical and practical feasibility of such building code visualization systems.

Procedure

One researcher conducted interview sessions remotely over Zoom, video recorded, and transcribed them. They took notes during the interview to identify key themes. Participants were briefed that they were going to be presented with a set of low-fidelity conceptual designs through a set of scenarios to illustrate their use. Participants were invited to imagine using the tool and to think creatively about potential applications of proposed technologies, how designs might be improved, any forthcoming challenges, and how such challenges might be addressed. Following each scenario, the interviewer drew on a set of structured questions to prompt discussion: *"Would you use the proposed features, why or why not?", "Do you see any other applications for the proposed technology?", "Would you trust the system?".*

Analysis

One researcher reviewed notes from interviews and transcripts, extracted quotes, and discussed central themes with all authors. The researcher developed a code structure closely following the interview guide questions and applied it to transcripts in a single pass. In addition, they extracted video frames where participants annotated images.

Building Code Visualization

Design. Our low-fidelity prototype (**Fig. 2**) was informed by input from the interviews and the survey, suggesting that real-time visualization of building codes would be most useful in early stages of design involving spatial building codes such as those regulating fire safety. We used the International Building Code [4], common in the USA and several other countries to guide our design. Very similar variations of such codes are common internationally. Our probe explores how showing the "available space" relative to these building code constraints could aid the design process. Finally, we represent alternative design goals that often present trade-offs with building codes by incorporating a program area, a set of spatial specifications that clients give to architects to satisfy.

We invited architects to imagine themselves iteratively developing alternative versions of a floorplan for a school while optimizing trade-offs between building codes regulating dimensions and arrangements of rooms and a client's requested room sizes. We explored the concept of showing "available space" by representing the relative discrepancy between critical thresholds in building codes, the associated spatial dimensions of a building, and the various dependencies between derived code measurements. The size and intended use of rooms determine an estimated occupant load for each room, which is then used to calculate the number of occupants passages need to support along a common path to an exit. The size of each room is shown in bar charts with the corresponding occupant load represented using aligned strip charts (Fig. 2.4). The spacing of strips represents the functional relationship between size and occupants. Colour fill is used to differentiate between current occupants and potential occupants. Green lines represent the client's requested room size (Fig. 2.6), while door icons aligned to strip charts represent room size thresholds at which additional doors would need to be added to support the flow of occupants into corridors (Fig. 2.5). Finally, the arrangement of rooms is used to define a common path of travel to the exit along which a running sum of occupants is used to calculate minimum corridor widths. These are represented using blue rectangle overlays (Fig. 2.2) and bar charts (Fig. 2.7).

In addition, inspired by our discussions with architects as well as adjacent visualization research using difference highlighting to compare alternatives in generative design [3], we introduced a lightweight on-the-fly versioning system that tracks and highlights the impact of design decisions (Fig. 2.8) through comparisons to the previous version (Fig. 2.9). This feature is intended to support design exploration by allowing designers to navigate between versions of their designs and explore the impact of intermediate design decisions.

Findings. Participants had generally favourable impressions of the proposed use of visualization and discussed how tool features could support **design**

This article has been accepted for publication in IEEE Computer Graphics and Applications. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MCG.2023.3307971

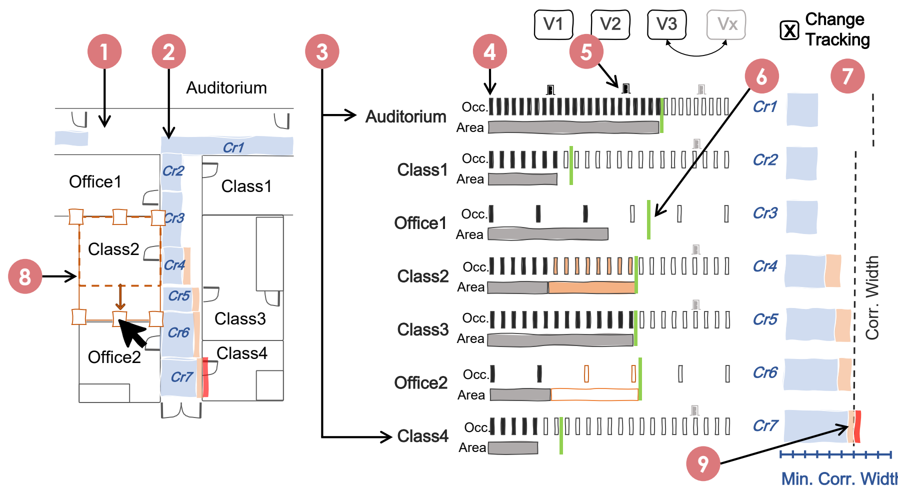

FIGURE 2. Mock-up of concept system to visualize building codes: (1) Floorplan within a CAD program. (2) Minimum corridor width overlays determined by estimated occupants. (3) Room use types. (4) Occupant load calculations shown by room size bar chart and corresponding strip chart showing people. (5) Additional icon markup showing thresholds such as additional doors. (6) Green bars representing client-requested room sizes. (7) Minimum widths of corridor segments determined by occupant load flowing from each room into the path of escape. (8) Change tracking while editing CAD model. (9) Highlighting changes in orange and compliance issues in red relative to previous version compliance issues.

exploration. They reflected on how the representation of "available space" would help them anticipate the impact of design decisions without having to make those decisions. *"I can kind of see that visually . . . here is where it becomes non-compliant. . . . it will be a conscious decision." (P6).* They also discussed how the change-tracking feature could support lightweight what-if style analyses (P12, P14, P8, P6), facilitate discussions or negotiations with authorities (P8), and serve as a visual reminder of past design decisions that help to orient exploration. *"[We think about] why it's like this . . . why we're doing this. Always relational. [That's why] that kind of tracking and that history . . . becomes more valuable." (P6).*

The participants appreciated the **legibility** visualizations provided. P5 appreciated having *"visual cues"* to help them *"remember how [building code logic] works . . . without having to [look it up]"*, while P3 expressed a need for more effective visual metaphors to reinforce understanding. *"They maybe could be more graphically clear [by representing] flow or directionality."*

Participants wanted more **direct interactions** with underlying data. They wanted to use visualizations to directly manipulate the drawing and configure building code rules. Rather than simply using visualizations as feedback evaluating the current state of a design P14 suggested using bar charts to manipulate the sizes of rooms to optimize across goals and constraints before re-arranging them in the adjacent CAD canvas space. *"You could do it the other way . . . the user could move [the room size bar and] resize the proportions of the classroom"*. Meanwhile, P3 said they *"would expect a tool like this to [allow them to] choose"* building code rule configurations interactively. Participants wanted to directly manipulate underlying data mechanics rather than just the representations.

Finally, participants raised **challenges of scaling** visualizations to larger architectural drawings and broader sets of building codes. To address these challenges, they suggested applying the existing structure of building code documentation (P13, P14) or the building model (P6, P14) to organize informational displays. P14 discussed how the concept of "available space" could be abstracted to provide aggregated numerical summaries in more complex displays and how it would support design decisions. *"You can get an . . . overall view of information, then you can narrow it down to very specific areas of the dataset by clicking portions of the graphic . . . I like this option, and then that option . . . and then you can see the ripple effect."* Participants also suggested in-situ visualizations such as our corridor width overlays (Fig. 1.2) will require clutter management systems similar to those already used in information-dense CAD tools (P13, P8, P6, P8, P10).

This article has been accepted for publication in IEEE Computer Graphics and Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MCG.2023.3307971

User-Driven Knowledge Management

Design. We presented a variety of mock-ups of workflows and features in an imagined CAD or BIM environment to explore a variety of user-driven knowledge mechanisms (**Fig. 3**) that could help address the challenges of complexity and ambiguity and make building code visualizations feasible. Our designs draw inspiration from existing work practices architects shared with us and attempt to streamline the fragmentation that is commonly involved. We used these mockups to delve into deeper issues of control, trust, and technical feasibility, inviting architects to participate creatively by imagining alternative implementations and applications. We presented these through illustrative scenarios based on the interview and survey findings.

In one scenario, we presented an imagined workflow as a prompt for discussions about the ways architects could have more supervisory oversight, control, and transparency when dealing with computed code rules. During interviews, architects described a system analogous to a spell-checker applied to a subset of context-relevant building codes. We incorporate this functionality using compliance highlighting ghost overlay (Fig. 3.1) and a checklist of context-relevant building codes (Fig. 3.2). A way to configure codes with links to source materials (Fig. 3.3), and a change history of building code configurations with commenting (Fig. 3.4) is inspired by the understanding that codes are dynamic, in conflict, often project-specific, and that architects rely on collaborative documentation to set project requirements based on building codes. We used this scenario to probe architects about issues of control, trust, and mechanisms to allow supervisory control over automation.

Our second scenario is inspired by the understanding that architects spend a considerable amount of time marking up and discussing architectural drawings and referencing specific relevant building code clauses. Our design explores ways to reduce fragmentation in these processes by allowing markup, including comments and references to building codes, to be computationally linked and anchored to the CAD design (Fig 3.5). We invited participants to imagine how they might use this functionality and what types of information would be worthwhile to embed in this way. We used this scenario to probe architects about the types of knowledge they might wish to embed, the collaborative purposes this system could serve, and any other needed functionalities.

Findings. Participants thought that **linking and annotating** CAD models with building code references would facilitate collaboration and coordination, streamline documentation, and help to disambiguate building code requirements. *“Popping up reference documents . . . that’s exactly what I would want to see in that situation.” (P15).* P8 imagined how this functionality could facilitate discussion and requirements gathering in meetings with authorities *“[You could] embed notes from a compliance officer . . . it was all tied to the geometry and you could go back in time, you can see what the conversation was referring to . . . you have this kind of like back and forth.”* Others saw applications for coordinating collaborations with other architects and professionals, particularly principal architects responsible for checking and marking up the work of their juniors (P2, P6, P8, P3, P5). P14 and P12 discussed how this would be key for disambiguating interpretations and maintaining exceptional code applications. *“[you need 5ft halls everywhere else] but in this scenario, you’re allowed to get away with a 4ft one, it’s good just to have that linked . . . and not have to be something that you remember.”*

Participants stressed issues of **trust and transparency** around automated compliance checking in the mockup we presented. *“If they find sometimes . . . it just screwed up, then it’s going to erode confidence.”* (P10). P8 explained that they would not fully trust such a system because many elements of the building are not likely to be perfectly modelled. *“Do you know the profile of the sink? . . . I would be very . . . reluctant to trust something at this resolution.”* Meanwhile, P5 and P3 explained that, as the legally liable party, an architect will always need to check all measurements themselves and would not fully rely on an automated system but could use it to facilitate their work if it did not create additional overhead. *“Software can’t assume that responsibility, but what it can do is help and help the person discover conditions that are non-compliant and help them remember how they reached compliance.”*

P14 reflected how annotations linked to a CAD model could increase trust by allowing collaborators to see the rationale and logic of how a building code was applied and interpreted *“as you progress during analysis, you may want teammates to understand the logic of your decision by having a traceable history of your thought process.”*

Our design probes prompted discussion around mechanisms for architects to themselves **implement building code** rules in a CAD system. Participants suggested that the existing visual and interaction language of CAD or BIM environments could be leveraged to implement and apply code rules as needed. P12, P5, P10, and P3 suggested measurement tools currently used for repeated manual recalculation could

This article has been accepted for publication in IEEE Computer Graphics and Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MCG.2023.3307971

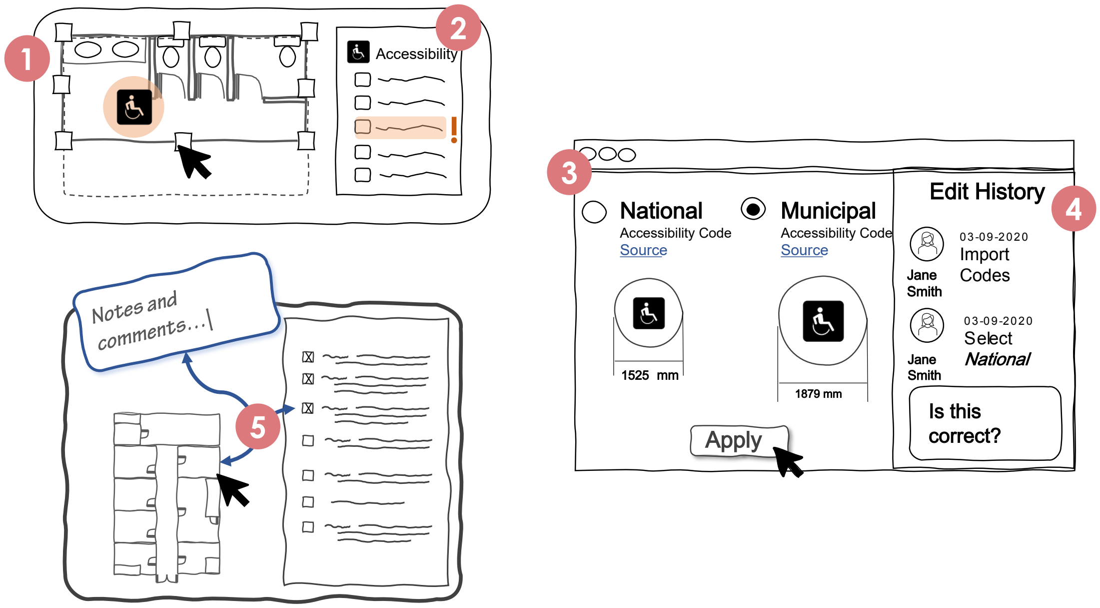

FIGURE 3. A series of mock-ups illustrating a user-driven knowledge management system: (1) Automated compliance checking. (2) Context-relevant building codes. (3) Building code configuration and access to source documentation. (4) Change history and commenting. (5) Annotations linking building codes with CAD models.

be extended such that they maintain distance measurements even as building models are reshaped and altered. P2 discussed how building elements would require specific BIM meta-data designations to enable the system to compute code logic. *"I think I need to designate . . . which doors are egress doors."* P15, P3, and P6 discussed how common calculations found in various parts of building codes could be abstracted to specific functions and embedded in an intuitive UI. Some architects expressed skepticism about whether the value such systems create would be worth the additional overhead of manual data entry and configuration. *"There's a lot of things that are sold to us about, you know, quicker modelling . . . a lot of the time . . . it causes more headaches than what it's worth." (P12).*

LIMITATIONS

As our research is qualitative and exploratory, it is limited in scope and not intended to generalize but rather to gather diverse perspectives, identify areas for future investigation, and highlight areas of transferability. While our participant pool represented a diverse range of countries, it was not comprehensive and was biased toward Western countries. Nevertheless, architects raised similar concerns and did not identify differences between countries as a distinct issue.

Our design probe study provoked reflections about how architects might see themselves using a tool rather than actually using it. While future formative research investigations should focus on the iterative development and evaluation of interactive prototypes, they will be subject to the same issues of ecological validity as our crude mockups. This is a common issue in visualization research, particularly in situations where tasks are ill-defined or emergent. Consequently, any summative evaluation of techniques or tools proposed in our work will need to be evaluated through ecologically valid real-world deployments.

DISCUSSION

Our findings corroborate the challenges of ambiguity and the neglected needs of architects [2]. While automated compliance checking aims to comprehensively address building codes and remove humans from the process, we find that human input and oversight are critical. Code interpretation and application are determined through social processes, building codes are part of design considerations, and architects are

This article has been accepted for publication in IEEE Computer Graphics and Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MCG.2023.3307971

liable for the safety of their designs, requiring them to personally ensure their drawings are compliant.

We propose this research suggests how CAD and BIM can be extended with visualizations and the concomitant user-driven knowledge management systems to better address the needs of architects using building codes as a design input. First, we add to literature at the intersection of architectural parametric design and visual analytics describing how visualization can aid design sensemaking, processes of creative exploration, decision-making, and speculation [3], [10], [12]. In addition, we add to a growing body of literature arguing that conventional visualization approaches do not adequately address the challenges of ambiguous and complex data because they treat the analysis and manipulation of representations of data as separate from the analysis and manipulation of data itself [15], [16]. The need for transparency, legibility, agency and direct interactions with underlying data characterizes the agenda of the emerging field of Human-Data Interaction (HDI). *" ...human manipulation, analysis and sensemaking of large, unstructured and complex datasets ... It addresses ... transparency, trust [direct] interaction..."* [16].

Our findings suggest that visualization could play a beneficial role in aiding architects in design explorations at early design stages. However, because building codes are complex and ambiguous, purely automated solutions will likely be inadequate. To be effective, any building code visualizations that are integrated into an architectural CAD or BIM program will need to include mechanisms that allow users to manage code rules based on project-specific requirements. These mechanisms should also facilitate the social processes involved in interpreting and documenting building code applications. Finally, the system should be designed so that architects can trust that it is accurate and reliable without having to spend extra time checking its work. An overarching theme throughout our findings is the need for transparency, legibility, and direct interactions with the data. In order for architects to use and trust building code visualization as design input, it is important that they have an understanding of the underlying data processing mechanisms and the ability to configure them as necessary. We make specific design recommendations based on this understanding below.

Show the design space, not just issues

Instead of solely displaying compliance issues, visualization designs should focus on revealing the design space within constraints, enabling architects to explore alternative solutions. This approach eases the burden of manually recalculating constraints, promotes trust through transparency and agency, and enhances the creative design process by helping architects anticipate the implications of design decisions without having to execute them.

Repurpose familiar tools for implementing building code logic

To reduce the overhead of manually implementing building code logic, extend and tailor existing CAD and BIM measurement tools to allow architects to specify and apply code rules as needed for a project, and then let the system automatically recalculate constraints as designs change. This approach reduces overhead in the compliance process overall, but in contrast to purely automated approaches, does so by empowering architects with transparency and control over code interpretation and application. Extending existing and familiar tools to specifically target building codes would minimize the learning curve to gain proficiency and avoids the fragmentation that navigating outside the familiar CAD workspace would introduce.

Embed knowledge & documentation in the workspace

Given the dynamic and collaborative nature of code interpretation and application, relevant reference materials should be embedded within the CAD or BIM model for easy retrieval. Having a record of design requirements, discussions shaping those requirements, references to specific legal language, and any exceptional circumstances is critical for the design process but can be difficult to navigate. Embedding such information to the CAD or BIM model so that it is retrievable in place will reduce the burden of finding such reference materials.

CONCLUSION

Our study demonstrates opportunities for interactive visualizations to support architectural design exploration using building codes, particularly at early design stages. Such tools could help architects maintain awareness of code constraints, explore design options within constraints, and anticipate the consequences of potential design decisions. However, the success of such tools critically depends on how well they address the complexity, ambiguity, and social processes of interpreting and applying building codes without creating additional overhead. We advocate for transparent, legible, and configurable designs to ensure architects maintain control and can trust system output. Future work should focus on integrating building code visualizations and knowledge management systems within existing CAD and BIM software and evaluating their effectiveness in practice.

ACKNOWLEDGMENTS

We thank all the professional architects who offered their time and insights. This research was supported in part by the National Sciences and Engineering Research Council of Canada (NSERC) under Grant IRCPJ 545100 - 18.

REFERENCES

1. R. Amor and J. Dimyadi, “The Promise of Automated Compliance Checking,” *Developments in the Built Environment*, vol. 5, p. 100039, 2021, doi: 10.1016/j.dibe.2020.100039 (Journal)
2. J. Soliman-Junior, P. Tzortzopoulos, and M. Kagioglou, “Designers’ Perspective on the Use of Automation to Support Regulatory Compliance in Healthcare Building Projects,” *Construction Management and Economics*, vol. 40, no. 2, pp. 123–141, 2022, doi: 10.1080/01446193.2021.2022176 (Journal)
3. L. Zaman et al., “GEM-NI: A System for Creating and Managing Alternatives In Generative Design,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* , pp. 1201–1210, 2015. (Conference proceedings)
4. F. D. Ching and S. R. Winkel, "Building Codes Illustrated: A Guide to Understanding the 2021 International Building Code." John Wiley & Sons, 2021. (Book)
5. N. Nawari, “A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking,” *Buildings*, vol. 9, no. 4, p. 86, 2019, doi: 10.3390/buildings9040086 (Journal)
6. S. Fenves, J. Garrett, Kiliccote, Law, and K. H. Reed, “Computer Representations of Design Standards and Building Codes: U.S. Perspective,” *The International Journal of Construction* IT3(1), pp.13-34, vol. 3, Jan. 1995. (Journal)
7. BOCA, Building Officials “Code Administration International Inc,” 1993. (Book)
8. S. Sobhkhiz, Y. C. Zhou, J. R. Lin, and T. E. El-Diraby, "Framing and Evaluating the Best Practices of IFC-based Automated Rule Checking: A Case Study. *Buildings*, 11(10), 456, 2021, doi: 10.3390/buildings11100456 (Journal)
9. J. S. Gero, “Design Prototypes: A Knowledge Representation Schema for Design,” *AI Magazine*, vol. 11, no. 4, pp. 26–26, 1990, doi: 10.1609/aimag.v11i4.854 (Journal)
10. R. F. Woodbury and A. L. Burrow, “Whither Design Space?,” *AI EDAM* , vol. 20, no. 2, pp. 63–82, 2006, doi: 10.1017/S0890060406060057 (Journal)
11. P. Ivson, A. Moreira, F. Queiroz, W. Santos, and W. Celes, “A Systematic Review of Visualization in Building Information Modeling,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 10, pp. 3109–3127, 2020, doi: 10.1109/TVCG.2019.2907583 (Journal)
12. H. Erhan, R. Sanchez, R. Woodbury, and V. Mueller, “Visual Narratives of Parametric Design History: Aha! Now I see how you did it!” in *Proceedings of the Education and Research in Computer Aided Architectural Design in Europe Conference*, pp. 259–268, 2012. (Conference proceedings)
13. T. Munzner, "A nested model for visualization design and validation," *IEEE Transactions on Visualization and Computer Graphics*, 15(6). 2009, doi: 10.1109/TVCG.2009.111 (Journal)
14. M. Tohidi, W. Buxton, R. Baecker, and A. Sellen, “User Sketches: A Quick, Inexpensive, and Effective Way to Elicit More Reflective User Feedback,” in *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles* , 2006. (Conference proceedings)
15. L. Bartram, M. Correll, and M. Tory, “Untidy Data: The Unreasonable Effectiveness of Tables,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 686–696, 2022, doi: 10.1109/TVCG.2021.3114830. (Journal)
16. E. Z. Victorelli and J. C. D. Reis, “Human-Data Interaction Design Guidelines for Visualization Systems,” in *Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems* . pp. 1–10, 2020. (Conference proceedings)
17. R. Xu, W. Solihin, and Z. Huang, “Code Checking and Visualization of an Architecture Design,” in *Proceedings of IEEE Visualization*, pp. 10p–10p. Oct. 2004, doi: 10.1109/VISUAL.2004.14. (Conference proceedings)