- Suggested Citation: Xue, X., Zhang, J., and El-Gohary, N. (2022). "Interactive visual representation of inter-connected requirements in building codes." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 1004-1012.
- Final Published Version can be found in ASCE Database here: https://ascelibrary.org/doi/10.1061/9780784483961.105

Interactive Visual Representation of Inter-Connected Requirements in Building Codes

Xiaorui Xue, S.M.ASCE,[1] Jiansong Zhang, Ph.D., A.M.ASCE[2*], and Nora El-Gohary, Ph.D., A.M.ASCE[3]

[1]Automation and Intelligent Construction (AutoIC) Laboratory, School of Construction Management Technology, Purdue University, West Lafayette, IN, 47907, PH (765) 430-2009. Email: xue39@purdue.edu.

[2]Automation and Intelligent Construction (AutoIC) Laboratory, School of Construction Management Technology, Purdue University, West Lafayette, IN, 47907, PH (765) 494-1574; FAX (765) 496-2246 (*corresponding author). ORCID: https://orcid.org/0000-0001-5225-5943. Email: zhan3062@purdue.edu.

[3]Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, Urbana, IL, 61801, PH (217) 333-6620. E-mail: gohary@illinois.edu

ABSTRACT

To facilitate a better understanding of building codes, the visualization of the embedded structures of the provisions and requirements of the codes is needed. Existing research efforts in building code compliance checking mostly do not purposefully represent building codes in formats that facilitate human understanding and interaction with the codes, such as XML and hypertext (text with links to other text). Visual programming commonly represents building codes more visually as flowcharts. However, flowcharts are static, and the generation of flowcharts is still manual. To address this lack of interactive visual representation of building code requirement structures, this paper proposes an automated building code structure extraction and visualization method for visualizing building code contents in a way that clearly shows the inter-connections between requirements and allows intuitive user interaction. In this method, to extract the chapter-section-subsection hierarchical structure and cross-reference structure, a new extraction method named Building Code Network Generator (BCNG) is proposed to automatically generate an interactive visualization using a directed network. The

- Suggested Citation: Xue, X., Zhang, J., and El-Gohary, N. (2022). "Interactive visual representation of inter-connected requirements in building codes." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 1004-1012.
- Final Published Version can be found in ASCE Database here: https://ascelibrary.org/doi/10.1061/9780784483961.105

performance of the proposed BCNG was empirically tested on Chapters 5 and 10 of the International Building Code 2015, with a resulting precision, recall, and F1-score of 99.4%, 96.3%, and 97.8%, respectively. In addition, the extracted hierarchical and cross-reference structures were displayed using an open-source network visualization tool to facilitate human understanding and interactions with the building code requirements in automated compliance checking systems.

BACKGROUND

Building codes are hard to read, even for experienced architecture, engineering, and construction (AEC) practitioners. A building code visualization method can help with the navigation and understanding of building codes. In addition, treating building code requirements as interconnected items instead of isolated ones allows for more holistic automated code compliance checking. Automated code compliance checking systems require building code requirements to be represented in a computer-processable format (Zhang and El-Gohary 2015). However, existing building code representations are limited in two ways. First, building code representations are limited in human readability. For example, Tan et al. (2010) developed a system of decision tables to automatically check the design of building envelopes. Although software can check a group of decision tables efficiently and accurately, human errors can be introduced when manually checking multiple decision tables simultaneously. Zhang and El-Gohary (2016) used B-Prolog logic rules to represent building codes. Although B-Prolog logic rules minimized ambiguity and supported a fully automated reasoning process, knowledge of logic is required to understand these rules. Second, building code representations in automated compliance checking systems did not explicitly represent connections between requirements. For example, Dimyadi et al. (2016) and Preidel and Borrmann (2017) represented building code requirement as flowcharts. Existing building code representations, although may function well, mostly treated requirements as separate items, with limited to no consideration of connections with other requirements. However, cross-references in building codes are abundant. To address this gap, in this research, the authors proposed a new rule-based relation extraction method, the Building Code Network Generator (BCNG), to extract and visualize hierarchical and cross-reference structures in building codes. The extraction of cross-references in building codes supports processing of connections between building code requirements. Visualization of the hierarchical and cross-reference structures could improve the readability of building code representations.

- Suggested Citation: Xue, X., Zhang, J., and El-Gohary, N. (2022). "Interactive visual representation of inter-connected requirements in building codes." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 1004-1012.
- Final Published Version can be found in ASCE Database here: https://ascelibrary.org/doi/10.1061/9780784483961.105

Visual programming in architectural, engineering, and construction. Visual programming aims to improve human programming experience by visual notations and visualization systems (Blackwell et al. 2001). By combining visual and text notation, visual programming smooths the steep learning curve of computer programming (Boshernitsan and Downes 2004). Application areas of visual programing language are numerous, including education (Tsai 2019), robotics (Coronado et al. 2020), geographic information system (GIS) (Calì et al. 2011), and Internet of Things (IoT) (Bak et al. 2020). Many research efforts explored the potential utilization of visual programming in the AEC domain as well. For example, Ghannad et al. (2019) used visual programming language (VPL) to validate building information modeling (BIM) data automatically. Zhu et al. (2020) combined BIM and VPL in a segment design method and tested their proposed method in a tunnel design. Preidel and Borrmann (2017) developed VPL for automated code compliance checking in general. In contrast to Preidel and Borrmann's VPL which needed to be generated manually and was mainly for rule execution, there is a need of visualization that can be generated automatically and used for code requirement understanding and navigation.

Relation extraction. Relation extraction is a natural language processing task that extracts sematic relationships between entities from natural language text (Bach and Badaskar 2007). Relation extraction algorithms use features such as the entities themselves, surrounding words of the entities, and dependency trees. Because manual feature engineering is difficult, kernels were used to search well-performing feature combinations automatically, which gave birth to feature-based relation extraction algorithms (Bunescu and Mooney 2005; Culotta and Sorensen 2004). The development of machine learning further gave birth to deep learning-based relation extraction (Zhang and El-Gohary 2021; Liu and El-Gohary 2021; Lv et al. 2016; Nguyen and Verspoor 2019; Qin et al. 2018).

METHODOLOGY AND EXPERIMENT

Building codes organize regulatory requirements in a hierarchical structure. One building code can be comprised of multiple chapters, each of which is devoted to one aspect of requirements such as: (1) general building heights and areas, (2) fire protection systems, and (3) means of egress. Building code chapters are further divided into sections and subsections. For example, Chapter 10 of the International Building Code (IBC) 2015 (International Code Council 2015) contains 30 sections and one section (e.g., Section 1010.1.10.1) of the chapter could have as many as four levels of subsections (Figure 1). This chapter-section-subsection hierarchy is evident and apparent. However, building codes also contain many cross-references, which are not explicated in the indexes or tables of content. One section of building codes is likely to refer to other sections for complementary information. For example, Section 1010.1.9.9 of the IBC 2015 refers to Section 1010.1.10 of the same code for information about *panic and fire exit hardware*. Building codes have many types of cross-references. One section of building code can refer to: (1) other sections in the same chapter, (2) other chapters, and/or (3) tables, equations, and terminologies in the code. To address the research gap in representing building code requirements as connected items, the authors proposed the Building Code Network Generator (BCNG), which is a rule-based relation extraction method that takes a three-step approach to extract hierarchical and cross-reference structures in the input building code and represent them in a network. BCNG stores the extracted network elements as a list of edges. The extraction steps are: regulatory information entity identification, hierarchical structure extraction, and cross-reference structure extraction. Regulatory information entities are pieces of the building code that has a title or a heading, such as a section, a chapter, or a table. The three steps are executed in sequence. When a regulatory information entity is identified, its linkages to other regulatory information entities in the hierarchical structure and its references to other regulatory information entities are extracted. The visualization component of BCNG can render the extracted network in 2D and 3D.

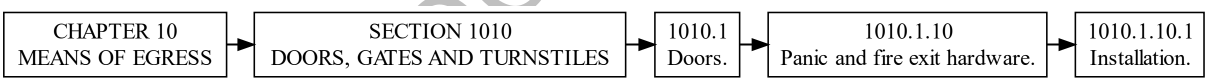

Figure 1. Sample hierarchy.

Regulatory information entities are indicated by titles of a segment of the building code. For example, the title of a section or a table indicates a regulatory information entity. Regulatory information entities can refer to other entities and be referenced by other entities. BCNG supports five types of entities: chapter, section, table, equation, and terminology.

The extraction of the hierarchical structure aims to extract all sections and subsections of a chapter in a tree data structure. The root of a tree is the title of the chapter. The child nodes of the root nodes are the sections of the chapter. The child nodes of a section node are the subsections of the section. A subsection node can have further child nodes if the subsection has its own subsections (i.e., subsubsections, etc.). The leaf nodes of the tree are subsections that do not have further subsections. In the extraction of the hierarchical structure, BCNG creates edges, where the sources are the parent nodes of the tree and the targets are the child nodes of the tree. A sub-tree of the hierarchical structure of Section 1022 of Chapter 10 of the IBC 2015 is shown in Figure 2.

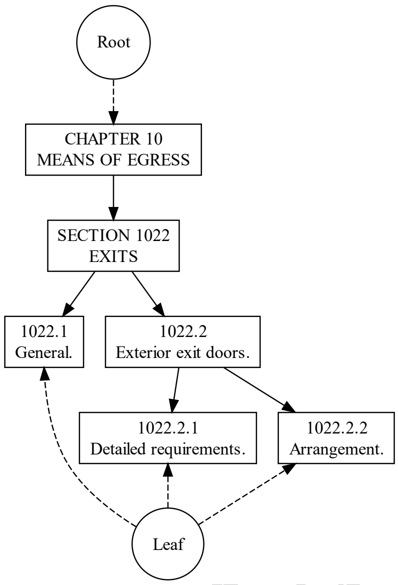

Figure 2. Sample tree.

BCNG utilizes pattern-matching rules to extract cross-references in the building code by creating edges that connect sources and targets of cross-references. It supports the extraction of references of a section or subsection of the building code to tables, equations, chapters, and other sections or subsections. And it renders the extracted networks in 2D and 3D. In the 2D visualization, names of linked regulatory information entities are connected by lines. In the 3D visualization, regulatory information entities are represented by colored spheres, and the names of the entities are annotated nearby. The visualization component of BCNG operates in a cloud server. After that, the visualized networks are rendered in a web page, which can be interacted with via a web browser. BCNG supports multiple interactions, such as zoom in, zoom out, pan, and rotate.

Dataset preparation. To test the proposed BCNG, the authors experimentally implemented it on a selected building code sample, Chapters 5 and 10 of the IBC 2015. These two chapters were selected because they have wide applicability by their provisions related to building area, height, and means of egress. Chapter 5 is about building area and building height. Chapter 10 is about means of egress. The hierarchical and cross-reference structures of these two chapters were extracted and visualized. A gold standard, which contains the hierarchical and cross-reference structures of these two chapters, was manually developed in the experiment for evaluation purposes.

Rule development. Because building codes are drafted in different ways by different authorities, BCNG requires customized rules for different building codes. The authors developed customized rules for the sample building codes in the implementation of BCNG. To ensure the scalability and effectiveness of the extraction rules, the authors developed the following guidelines for rule development: (1) one extraction rule should only extract one type of hierarchical or cross-reference structure, (2) extraction rules cannot alter results of other rules, and (3) extraction rules cannot introduce errors. Guideline #1 means if one rule is designed to extract references to equations, it cannot be used to extract references to other contents such as tables. Guideline #2 means if a reference is extracted, it is final. Other rules cannot delete or modify existing extracted references. Guidelines #1 and #2 give BCNG the ability to cover new types of cross-references simply by adding new rules. Guideline #3 requires the extraction results of a rule to be checked immediately after the rule is added. Guideline #3 ensures the accuracy of extraction. The development of extraction rules follows a data-driven and iterative approach (Figure 3). First, a gold standard of the hierarchical structure and the cross-reference structure of the sample building code is developed. After that, extraction rules are developed to extract the hierarchical structure. Finally, extraction rules that extract cross-references are developed. The extraction results of the rules were iteratively checked against the training dataset after each modification of extraction rules to ensure the accuracy of extraction in the rule development.

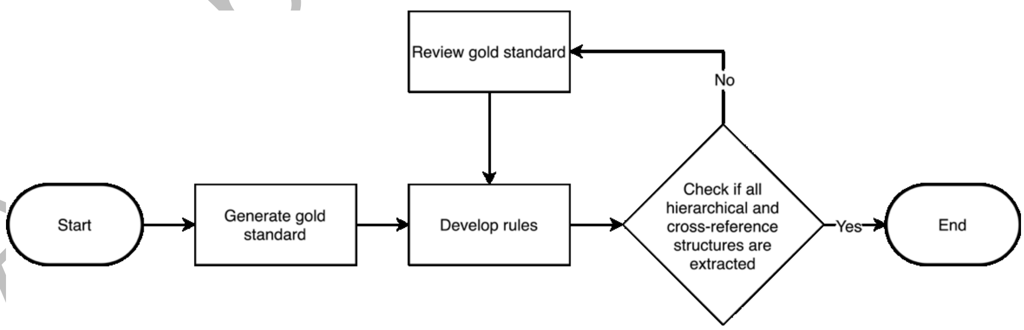

Figure 3. Rule development procedure.

Regulatory information entity identification rules. BCNG extracts five types of regulatory information entities: chapter, section, table, equation, and terminology. It uses keywords and

- Suggested Citation: Xue, X., Zhang, J., and El-Gohary, N. (2022). "Interactive visual representation of inter-connected requirements in building codes." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 1004-1012.
- Final Published Version can be found in ASCE Database here: https://ascelibrary.org/doi/10.1061/9780784483961.105

rules to extract regulatory information entities. A chapter starts with the keyword *CHAPTER*. Sections are mentioned in two ways: (1) indicated by keyword *SECTION* or *Section*, and (2) indicated by section numbers. Section numbers are numerical references to sections, which were entirely composed of numbers and dots. Sections numbers also need to be at least three characters long and should not end with a dot. This constraint excludes list marks and other listing numbers. Tables are indicated by keywords *Table(s)*. Equations are indicated by the keyword *Equation*. Terminologies are listed in the terminology sections of building codes.

Hierarchical structure extraction rules. The hierarchy of a building code can be represented as a tree structure, which includes all sections and sub-sections of the building code. The extraction of building code hierarchical structure follows a top-down approach. First, root node begins with the keyword *Chapter*. Second, the child nodes of the root node are the section nodes that begin with the keyword *Section*. Third, the child nodes of a section node are its subsections. For example, the child nodes of Section 1022 are its subsections, namely, Sections 1022.1 and 1022.2.

Cross-reference extraction rules. Cross-references are described as edges, which start at the source (i.e., a line of building code that refers to other parts of the building code) and end at the target (i.e., a regulatory information entity that is being referenced). BCNG requires sources and targets of cross-references to be the most specific section that a line of building code belongs to. For example, a line of building code of a chapter belongs to that chapter. Therefore, although it is correct to say the source reference is the chapter, this type of cross-references does not have much value. Instead, sources and targets of references need to be as specific as possible, i.e., at the lowest level of subsections.

BCNG implements the following rules to extract sources of cross-references. First, the most specific section that a line of building code belongs to is listed at the beginning of the line. Second, if a line is not associated with a section number, it belongs to the same section of the line before it. Target of cross-reference are defined as regulatory information entities that are mentioned in a section of building code. The identification of entities and extraction of cross-references are operated in parallel. Cross-references to a regulatory information entity are extracted when the entity is identified.

A gold standard of the embedded structure of the sample building code was developed by the authors prior to the development of rules. The gold standard includes the entire hierarchy, and all cross-references of the sample building codes. First, all regulatory

- Suggested Citation: Xue, X., Zhang, J., and El-Gohary, N. (2022). "Interactive visual representation of inter-connected requirements in building codes." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 1004-1012.
- Final Published Version can be found in ASCE Database here: https://ascelibrary.org/doi/10.1061/9780784483961.105

information entities, except for bolded section titles (i.e., already highlighted by bold fonts) in the sample building code, were identified and highlighted. Second, the hierarchical and cross-reference structures of the sample building code were manually extracted. The regulatory information entities and edges were strictly defined. Therefore, the formation of gold standard was unambiguous. In total, the cross-reference network of Chapter 5 of the IBC 2015 contained 159 regulatory information entities and 318 edges, and the cross-reference network of Chapter 10 of the IBC 2015 contained 562 regulatory information entities and 970 edges.

Results. The results were compared against the gold standard automatically (by a computer program), which counted true positives (the BCNG extracts an edge that is in the gold standard), false positives (the BCNG extracts an edge that is not in the gold standard), and false negatives (the BCNG does not extract an edge that is in the gold standard) of the extraction. True negatives are not applicable to this study because true negative means the BCNG does not extract an edge that is not in the gold standard. The precision is defined as true positive / (true positive + false positive). The recall is defined as true positive / (true positive + false negative). The F1-score is defined as 2 * (recall * precision) / (recall + precision). BCNG reached 99.7%, 98.0%, and 98.8% precision, recall, and F1-score on Chapter 5 of the IBC 2015 and 99.4%, 95.8%, and 97.5% precision, recall, and F1-score on Chapter 10 of the IBC 2015. The overall precision, recall, and F1-score are 99.4%, 96.3%, and 97.8%, respectively.

The extracted networks of both chapters were visualized in both 2D and 3D. Figures 4 and 5 show the 2D and 3D visualizations of the hierarchical and cross-reference structures of Chapters 5 and 10, respectively. BCNG also supports the interaction with extracted networks by zooming in, zooming out, pan, and rotating, as illustrated in the figures.

- Suggested Citation: Xue, X., Zhang, J., and El-Gohary, N. (2022). "Interactive visual representation of inter-connected requirements in building codes." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 1004-1012.
- Final Published Version can be found in ASCE Database here: https://ascelibrary.org/doi/10.1061/9780784483961.105

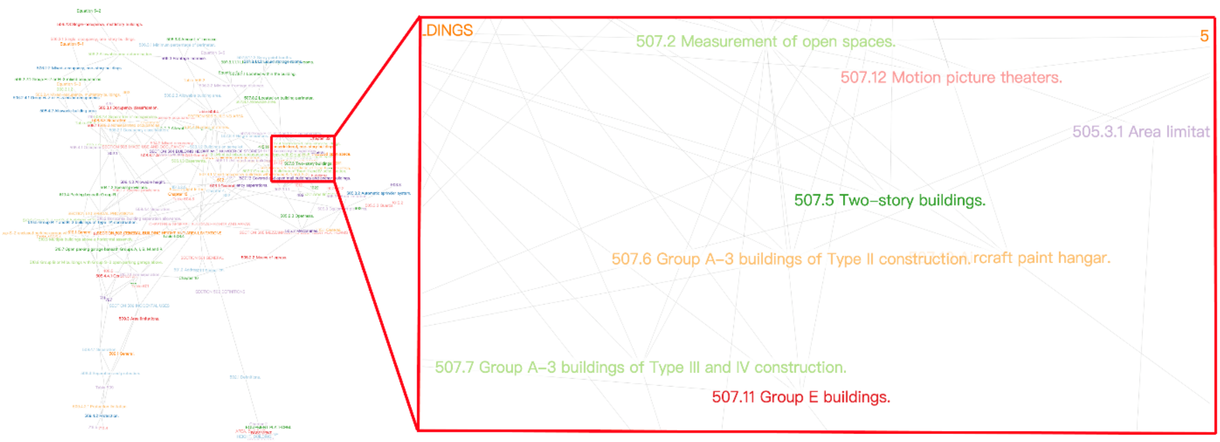

Figure 4. 2D visualizations of hierarchical and cross-reference networks of Chapter 5 of the International Building Code 2015.

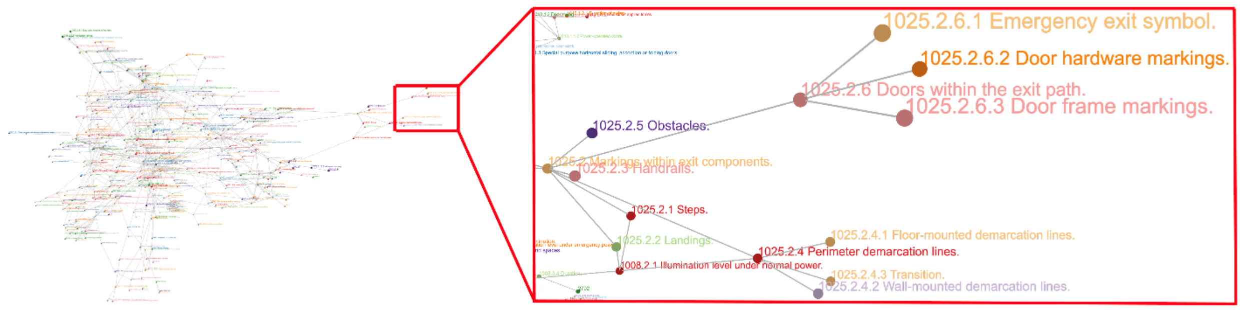

Figure 5. 3D visualizations of hierarchical and cross-reference networks of Chapter 10 of the International Building Code 2015.

CONCLUSION

This paper presented the BCNG, a rule-based relation extraction method, to extract hierarchical and cross-reference structures from building codes. Three extraction rule development guidelines were proposed and followed. An iterative extraction rule development method was also presented. Customized extraction rules were developed to extract hierarchical and cross-

- Suggested Citation: Xue, X., Zhang, J., and El-Gohary, N. (2022). "Interactive visual representation of inter-connected requirements in building codes." Proc., ASCE Construction Research Congress, ASCE, Reston, VA, 1004-1012.
- Final Published Version can be found in ASCE Database here: https://ascelibrary.org/doi/10.1061/9780784483961.105

reference structures of a sample building code. BCNG reached 99.4%, 96.3%, and 97.8% precision, recall, and F1-score on the extraction of the embedded structure of sample building code. The proposed interactive visualization allows users to interact with the extracted networks in a familiar and intuitive way to facilitate the understanding of and navigation through the building code requirements.

LIMITATIONS AND FUTURE WORK

Although BCNG can extract hierarchical and cross-reference structures in building codes, it is still limited in the types of regulatory information entities it can extract and the granularity of the extracted network. For example, BCNG cannot extract references to other codes yet. If a building code section lists exceptions to the current section, BCNG cannot distinguish cross-references in the main body of the section from the exceptions of the section yet. Future work could expand the BCNG coverage to increase granularity of extracted networks and extractable regulatory information entities.

ACKNOWLEDGEMENT

The authors would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grant No. 1827733. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

Bach, N., and Badaskar, S. (2007). "A review of relation extraction." *Liter. Rev. Lang. and Stat. II*, 2, 1-15.

Bak, N., Chang, B.-M., and Choi, K. (2020). "Smart Block: A visual block language and its programming environment for IoT." *J. Comput. Lang.*, 60, 100999.

Blackwell, A. F., Whitley, K. N., Good, J., and Petre, M. (2001). "Cognitive factors in programming with diagrams." *Artif. Intell. Rev.*, 15(1), 95-114.

Boshernitsan, M., and Downes, M. S. (2004). "Visual programming languages: A survey." EECS Department, University of California, Berkeley.

Bunescu, R., and Mooney, R. "A shortest path dependency kernel for relation extraction." *Proc., Human Language Technology Conference and Conference on Empirical Methods in Natural Language*, 724-731.

- **Suggested Citation: Xue, X., Zhang, J., and El-Gohary, N. (2022). "Interactive visual representation of inter-connected requirements in building codes." Proc., ASCE Construction Research Congress, ASCE, Reston, VA,** 1004-1012.
- **Final Published Version can be found in ASCE Database here: https://ascelibrary.org/doi/10.1061/9780784483961.105**

Calì, D., Condorelli, A., Papa, S., Rata, M., and Zagarella, L. (2011). "Improving intelligence through use of Natural Language Processing. A comparison between NLP interfaces and traditional visual GIS interfaces." *Procedia Comput. Sci.*, 5, 920-925.

Chang, S.-K. (1987). "Visual languages: A tutorial and survey." *IEEE Software*, 4(1), 29-39.

Coronado, E., Mastrogiovanni, F., Indurkhya, B., and Venture, G. (2020). "Visual programming environments for end-user development of intelligent and social robots, a systematic review." *J. Comput. Lang.*, 58, 100970.

Culotta, A., and Sorensen, J. "Dependency tree kernels for relation extraction." *Proc., The 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 423-429.

Dimyadi, J., Clifton, G., Spearpoint, M., and Amor, R. (2016). "computerizing regulatory knowledge for building engineering design." *J. Comput. Civ. Eng.*, C4016001.

Ghannad, P., Lee, Y.-C., Dimyadi, J., and Solihin, W. (2019). "Automated BIM data validation integrating open-standard schema with visual programming language." *Adv. Eng. Inform.*, 40, 14-28.

International Code Council. (2015). *International building code*, ICC.

Liu, K., and El-Gohary, N. (2021). "Semantic neural network ensemble for automated dependency relation extraction from bridge inspection reports." *J. Comput. Civ. Eng.*, 35(4), 04021007.

Lv, X., Guan, Y., Yang, J., and Wu, J. (2016). "Clinical relation extraction with deep learning." *Int. J. Hybrid Inf. Technol.*, 9(7), 237-248.

Nguyen, D. Q., and Verspoor, K. "End-to-end neural relation extraction using deep biaffine attention." *Proc., European Conference on Information Retrieval*, Springer, 729-738.

Preidel, C., and Borrmann, A. (2017). "Refinement of the visual code checking language for an automated checking of building information models regarding applicable regulations." *J. Comput. Civ. Eng., 2017*, 157-165.

Qin, P., Xu, W., and Wang, W. Y. (2018). "Robust distant supervision relation extraction via deep reinforcement learning." *arXiv preprint arXiv:1805.09927*.

Reitsma, R. F., Sautins, A. M., and Wehrend, S. C. (1994). "Construction kit for visual programming of river & basin models." *J. Comput. Civ. Eng.*, 8(3), 378-384.

Tan, X., Hammad, A., and Fazio, P. (2010). "Automated code compliance checking for building envelope design." *J. Comput. Civ. Eng.*, 24(2), 203-211.

Tsai, C.-Y. (2019). "Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy." *Comput. Hum. Behav.*, 95, 224-232.

Zhang, J., and El-Gohary, N. M. (2015). "Automated information transformation for automated regulatory compliance checking in construction." *J. Comput. Civ. Eng.*, 29(4), B4015001.

Zhang, J., and El-Gohary, N. M. (2016). "Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking." *J. Comput.*

Civ. Eng., 30(2), 04015014.

Zhang, R., and El-Gohary, N. (2021). "A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking." *Autom. Constr.*, in press.

Zhu, X., Bao, T., Yu, H., and Zhao, J. (2020). "Utilizing Building Information Modeling and Visual Programming for Segment Design and Composition." *J. Comput. Civ. Eng.*, 34(4), 04020024.