Real-Time XR Visualizations of Code Metrics in the IDE

David Moreno-Lumbreras*, Gregorio Robles*, Adrian Montes-Linares*
* Escuela De Ingeniería de Fuenlabrada, Universidad Rey Juan Carlos, Fuenlabrada, Madrid, Spain
Emails: {david.moreno, gregorio.robles}@urjc.es, a.montesl.2020@alumnos.urjc.es

***Abstract*—As software grows in size and complexity, developers struggle to understand structural and metric relationships. We present *CodeXR*, a lightweight Visual Studio Code plugin that extends the IDE with real-time eXtended Reality (XR) visualizations of software metrics. By mapping properties such as lines of code, parameters, and cyclomatic complexity to 3D metaphors, developers can explore entire projects as interactive cities and monitor changes live as they code. CodeXR integrates directly with the IDE, updating visualizations continuously without requiring external dashboards. Additionally, it offers an optional XR representation of HTML DOM hierarchies to support web development tasks. Built on BabiaXR and leveraging concepts validated in prior work like CodeCity, this approach aims to enhance comprehension, reduce cognitive load, and promote spatial reasoning about code quality. We describe the design, capabilities, and use cases, and outline future directions to empirically evaluate its impact on developer workflows.**

***Index Terms*—extended reality, code metrics, software visualization, web**

I. Introduction

Understanding the structure and quality of software systems is a core challenge as projects grow in size and complexity [1]. Traditional integrated development environments (IDEs) provide tools for code analysis and visualization, often relying on flat 2D representations such as heatmaps, line charts, or tables [2]. Although these approaches are widely adopted in platforms like Visual Studio Code and GitHub, they can obscure relationships between metrics such as complexity, size, and dependencies, limiting developers' ability to detect structural issues intuitively [3].

To address these limitations, research has increasingly explored immersive techniques—particularly Virtual and Augmented Reality (VR/AR)—which can present software artifacts in spatial, multidimensional formats [4]. Foundational work has shown that spatial metaphors and embodied interaction can improve comprehension and engagement [5], and the city metaphor in tools like CodeCity [6] or ExplorViz [7] has become a reference model for visualizing code as 3D urban landscapes.

Building on this line of research, we present *CodeXR*, a tool that integrates real-time XR visualizations of software metrics directly into the IDE. By mapping metrics such as lines of code, parameter counts, and cyclomatic complexity to height, base area, and color, CodeXR helps developers track complexity trends and detect anomalies during development. Unlike prior standalone systems, it provides continuous feedback in AR or desktop 3D views without leaving the coding environment. Additionally, CodeXR offers optional XR visualization of HTML DOM hierarchies [8], supporting web development scenarios.

This work aims to explore how immersive, live metric representations can improve comprehension and potentially change how developers interact with code.

II. CodeXR

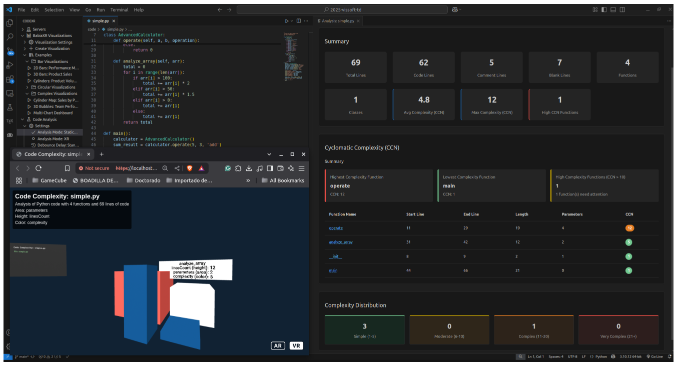
CodeXR is an IDE plugin built on BabiaXR [9], a web-based XR visualization toolkit using A-Frame. The system analyzes software metrics such as lines of code, parameter counts, and cyclomatic complexity via the Lizard engine. Each function or method becomes a building in a 3D city metaphor: height encodes lines of code, base area reflects parameters, and color gradient indicates complexity. This representation helps developers quickly detect hotspots, growth trends, and structural anomalies.

Internally, CodeXR launches a local HTTPS server to serve the visualization scene. As developers edit code in the IDE, changes are detected automatically and streamed in real time using Server-Sent Events (SSE), allowing continuous feedback without manual refresh. Developers can configure update intervals, color schemes, visualization styles, and metric thresholds through the plugin's dashboard.

CodeXR supports multiple languages including JavaScript, Python, C++, and Ruby. Developers can right-click any file or the entire project to launch the visualization in one of three modes:

- **Desktop 3D Mode:** A browser-based dashboard with interactive navigation, metric configuration, and filtering options (Figure 1).
- **AR Mode:** The city visualization is overlaid on the physical workspace using AR devices (e.g., Meta Quest), enabling immersive inspection while coding (Figure 2).
- **DOM Mode:** An optional XR view displays the HTML DOM hierarchy, supporting web development scenarios.

A typical workflow involves selecting a file or the entire repository to visualize module complexity and identify potentially problematic functions or classes. For example, during a Ruby refactoring session, duplicating logic and adding parameters caused the corresponding building in the scene to grow taller and darker, clearly indicating increased size and complexity. In larger projects, the visualization helps compare different modules and locate clusters of high-complexity code that may require attention.



```
class AdvancedCalculator:
    def operate(self, a, b, operation):
            else:
                return 0

    def analyze_array(self, arr):
        total = 0
        for i in range(len(arr)):
            if arr[i] > 100:
                total += arr[i] * 2
            elif arr[i] > 50:
                total += arr[i] * 1.5
            elif arr[i] > 0:
                total += arr[i]
            else:
                total += arr[i]
        return total

def main():
    calculator = AdvancedCalculator()
    sum_result = calculator.operate(5, 3, 'add')
```

Code Complexity: simple.py

Analysis of Python code with 4 functions and 69 lines of code

Area: parameters

Height: linesCount

Color: complexity

analyze_array
linesCount (height): 12
parameters (area): 2
complexity (color): 5

Summary

69	62	5	7	4
Total Lines	Code Lines	Comment Lines	Blank Lines	Functions
1	4.8	12	1	
Classes	Avg Complexity (CCN)	Max Complexity (CCN)	High CCN Functions	

Cyclomatic Complexity (CCN)

Summary

Highest Complexity Function
operate
CCN: 12

Lowest Complexity Function
main
CCN: 1

High Complexity Functions (CCN > 10)
1
1 function(s) need attention

Function Name	Start Line	End Line	Length	Parameters	CCN
operate	11	29	19	4	12
analyze_array	31	42	12	2	5
__init__	8	9	2	1	1
main	44	66	21	0	1

Complexity Distribution

3	0	1	0
Simple (1-5)	Moderate (6-10)	Complex (11-20)	Very Complex (21+)

Fig. 1: Desktop 3D Mode: interactive city visualization with live updates and configuration panel.

In contrast to prior tools like CodeCity [6], which run as standalone applications, CodeXR provides real-time integration inside the IDE, reducing context switching and enabling immediate feedback. The DOM mode further extends applicability by visualizing web document structures, making it easier to inspect nested elements and their relationships.

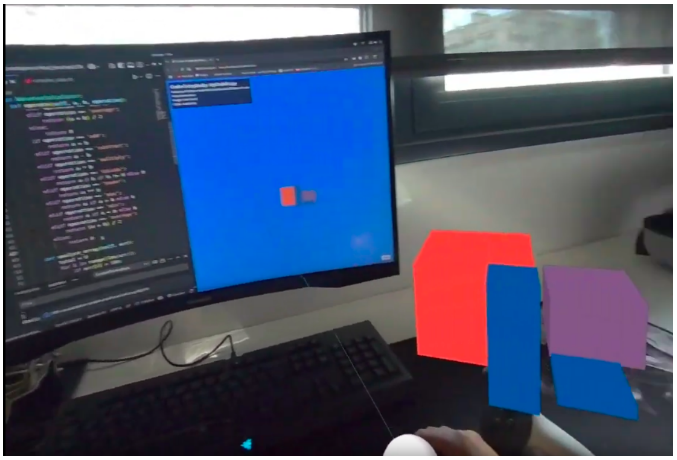


Fig. 2: AR Mode: real-time visualization of code metrics overlaid while coding.

This combination of live metric analysis, immersive exploration, and continuous updates aims to improve comprehension, reduce cognitive load, and foster better decision-making during development. More details and demonstrations are available in the replication package[1] and supplementary video[2].

III. Discussion and Conclusion

CodeXR demonstrates the feasibility of bringing real-time, immersive visualizations of software metrics and HTML structures into the IDE workflow. By combining a city metaphor with live updates, developers can immediately perceive how code evolves as they work, without switching between external tools or dashboards. This integration is designed to reduce cognitive overhead, support early detection of complexity growth, and encourage more informed design decisions.

Unlike prior approaches such as CodeCity [6] or ExplorViz [7], which require standalone environments, CodeXR embeds the visualization directly where developers spend most of their time. The DOM visualization capability builds on concepts validated in earlier work [8], further extending the tool's applicability to web development scenarios.

Although the system builds on established visualization paradigms, our prototype is still preliminary. So far, its design has been evaluated informally, and further empirical studies are needed to understand its impact on comprehension, productivity, and collaboration. Another limitation is scalability: while the tool can visualize entire repositories, rendering very large projects may require performance optimizations.

Future work will focus on refining the interaction model, integrating filtering and search capabilities, and enabling richer collaboration features—such as shared AR sessions for code reviews or pair programming. Additionally, as developers increasingly rely on AI-generated code, we believe spatial representations can play an important role in assessing the structural quality of machine-produced artifacts.

Ultimately, our vision is that tools like CodeXR will contribute to reimagining how developers engage with software systems: not only through text and static metrics but also through immersive, spatial perspectives that make complexity tangible and actionable.

[1] https://doi.org/10.5281/zenodo.15367992

[2] https://youtu.be/GFrGDt2BOW0

References

[1] R. Koschke, "Software visualization in software maintenance, reverse engineering, and re-engineering: a research survey," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 15, no. 2, pp. 87–109, 2003.

[2] T. Ball and S. G. Eick, "Software visualization in the large," *Computer*, vol. 29, no. 4, p. 33–43, apr 1996. [Online]. Available: https://doi.org/10.1109/2.488299

[3] B. Meyer, "Seven principles of software testing," *Computer*, vol. 41, no. 8, pp. 99–101, 2008.

[4] C. Demiralp, C. Jackson, D. Karelitz, S. Zhang, and D. Laidlaw, "Cave and fishtank virtual-reality displays: A qualitative and quantitative comparison," *IEEE transactions on visualization and computer graphics*, vol. 12, pp. 323–30, 05 2006.

[5] A. Batch, A. Cunningham, M. Cordeil, N. Elmqvist, T. Dwyer, B. H. Thomas, and K. Marriott, "There is no spoon: Evaluating performance, space use, and presence with expert domain users in immersive analytics," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 536–546, 2019.

[6] R. Wettel and M. Lanza, "Visualizing software systems as cities," in *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*, 2007, pp. 92–99.

[7] F. Fittkau, A. Krause, and W. Hasselbring, "Exploring software cities in virtual reality," in *2015 IEEE 3rd Working Conference on Software Visualization (VISSOFT)*, 2015, pp. 130–134.

[8] D. Moreno-Lumbreras, "Enhancing HTML Structure Comprehension: Real-Time 3D/XR Visualization of the DOM ," in *2024 IEEE Working Conference on Software Visualization (VISSOFT)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2024, pp. 127–132. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/VISSOFT64034.2024.00025

[9] D. Moreno-Lumbreras, J. M. Gonzalez-Barahona, and A. Villaverde, "BabiaXR: Virtual reality software data visualizations for the web," in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2022, pp. 71–74.